NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

CLASSIFICATION, SEARCH, AND RETRIEVAL IN A MULTI-VARIABLE, MULTI-LEVEL TAXONOMY: APPLICATION TO DECISIONNET

by

Christopher M. Corgnati

September, 1997

Thesis Advisor: Hemant K. Bhargava

Approved for public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE		Form Approved OMB No. 0704-0188
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.		
1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE September 1997	3. REPORT TYPE AND DATES COVERED Master's Thesis
4. TITLE AND SUBTITLE : CLASSIFICATION, SEARCH, AND RETRIEVAL IN A MULTI-VARIABLE , MULTI-LEVEL TAXONOMY: APPLICATION TO DECISIONNET		5. FUNDING NUMBERS
6. AUTHOR(S) Corgnati, Christopher M.		
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000		8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSORING / MONITORING AGENCY REPORT NUMBER
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.		
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution unlimited.		12b. DISTRIBUTION CODE

13. ABSTRACT *(maximum 200 words)*

The explosion of information available on global computer networks underlines the need for effective repositories that facilitate organization of, and search for, information. These digital repositories may contain simple data, or increasingly, objects of other types such as software and decision models. A taxonomy can be thought of as a navigational aid to a repository. Organization of objects may take place along multiple dimensions, each of which may have a taxonomy of classification terms that spans many levels.

This thesis examines the design and development of a WWW based Classification, Registration, Search, and Retrieval System. The system was applied and tested on the DecisionNet project which is an electronic brokerage house for decision technologies. In order to facilitate user interaction via the WWW the system was designed to be run through a standard web browser. A graphical user interface was developed in Java. The back-end functions for data management, search and retrieval were also programmed largely in Java.

14. SUBJECT TERMS Taxonomy, Repository, Information Search and Retrieval, Java Applet, DecisionNet, Decision Support Systems			15. NUMBER OF PAGES 168
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFI- CATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL

CLASSIFICATION, SEARCH, AND RETRIEVAL IN A MULTI-VARIABLE, MULTI-LEVEL TAXONOMY: APPLICATION TO DECISIONNET

Christopher M. Corgnati
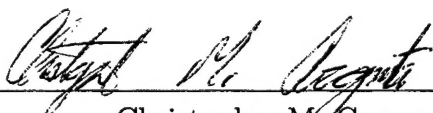Lieutenant, United States Navy
B.S., Villanova University, 1990

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN INFORMATION TECHNOLOGY MANAGEMENT

from the
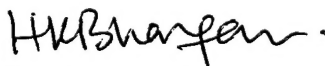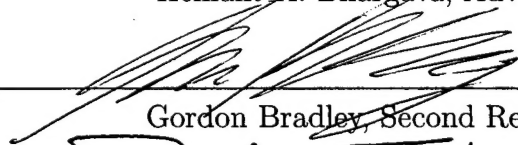
NAVAL POSTGRADUATE SCHOOL
September 1997

Author: _____
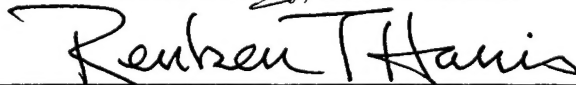
Christopher M. Corgnati

Approved by: _____

Hemant K. Bhargava, Advisor

_____

Gordon Bradley, Second Reader

_____

Rueben T. Harris, Chairman, Department of Systems
Management

ABSTRACT

The explosion of information available on global computer networks underlines the need for effective repositories that facilitate organization of, and search for, information. These digital repositories may contain simple data, or increasingly, objects of other types such as software and decision models. A taxonomy can be thought of as a navigational aid to a repository. Organization of objects may take place along multiple dimensions, each of which may have a taxonomy of classification terms that spans many levels.

This thesis examines the design and development of a WWW based Classification, Registration, Search, and Retrieval System. The system was applied and tested on the DecisionNet project which is an electronic brokerage house for decision technologies. In order to facilitate user interaction via the WWW the system was designed to be run through a standard web browser. A graphical user interface was developed in Java. The back-end functions for data management, search and retrieval were also programmed largely in Java.

v

TABLE OF CONTENTS

x

LIST OF FIGURES

I.   INTRODUCTION

A large number of World Wide Web (WWW) applications can be interpreted as providing access to a collection of digital resources such as information (e.g., The Library of Congress [Ref. 1]), or software code (e.g., The Java Repository [Ref. 2]), or mathematical decision models (e.g., DecisionNet [Ref. 3]). Collections or repositories, in the classical library science approach, are subject to operations such as classification and indexing, and search and retrieval. However, these operations are only weakly, if at all, supported in most Web-based repositories, leading to difficulties in locating suitable objects from large collections. For example, a search conducted using general search engines such as Excite [Ref. 4] or Lycos [Ref. 5] results in thousands of "hits" only a few of which may actually be relevant.

This thesis describes a system for classifying, registering, searching, and retrieving objects in a Web-based digital library, in a way that these functions can be accessed by users via their Web browsers. This work is done in the specific context of DecisionNet [Ref. 6], a digital library of decision technologies. However, in its architecture and design as well as in its implementation, the system is general enough to be immediately useful for classification and search in other repositories. Since the classification of objects in DecisionNet is done with multiple attributes, each of which has several levels, a Web-based user interface for access to classification and search functions poses significant challenges. Hence, much of the system and this thesis is focused on the development of a general, Web-based, user interface for a multi-variable, multi-dimensional classification of a repository of objects.

A.   THE DECISIONNET CONCEPT

DecisionNet is a framework for brokering the use of decision support technologies. Earley [Ref. 7] described DecisionNet in the following terms:

1

The main premise of DecisionNet is that consumers should be able to utilize decision support technologies as services over the World Wide Web (WWW), instead of purchasing them as stand-alone products. By providing decision support technologies over the WWW (either for free of for a nominal usage fee), users would be able to tap into the power of such technologies without the normally prohibitive costs involved with purchasing or downloading and installing a specific product. The DecisionNet prototype serves as an interface between consumers and providers of decision support software, so that interactions with multiple technologies may take place at one convenient location. In this sense, DecisionNet performs the role of an agent, facilitating transactions between consumers and providers of decision support technologies.

With that as a foundation we will explore the genesis of this thesis.

B. IMPROVING DECISIONNET

The need for an improved search engine in DecisionNet was first identified by Rogers. [Ref. 8] It stated, "Further research is needed on the second part, to develop intelligent search and retrieval algorithms." The search and retrieval problem continued and was further identified by Earley. [Ref. 7] It stated:

The indexed search script developed in this thesis assumes that the consumer is looking for exact matches to his query. However, there may be some technologies that are closely related to a consumer's query without being an exact match. The taxonomy for indexing technologies is already in place and functional. (Rogers, 1996) A method for assigning 'scores' based on a consumer's request would prove to be beneficial.

As the size of an object library grows, so will the diversity of the objects in that library. The taxonomy needed to accurately classify these diverse objects will also grow. A taxonomy is a shared set of terms that are relevant to the domain of knowledge that the objects are a part of. As the taxonomy and number of search criteria grow, the likelihood of finding exact matches drops very quickly. This is true of DecisionNet, which contains relatively few objects. These observations provide the basis for this thesis.

C. PROBLEM DEFINITION

Information retrieval on the WWW is both an art and a science. Keyword based searches may retrieve nothing or several million hits. The art is knowing what to search for. WWW search engines, even the "advanced" search engines, are notoriously inaccurate (commit both Type 1 errors and plenty of Type 2 errors). A Type 1 error occurs when a relevant result is ignored. A Type 2 error occurs when an irrelevant result is returned. A search for Apple Computers returns 58,312 hits or possible items of interest. Of those, approximately 320 are closely tied to the corporation in question. The remaining sites have little to no relevance to Apple Computers Inc. This example uses WWW sites, but the same idea applies in general to any 'object'. The challenge is to rank various objects giving the user the most relevant results. What is a good measure of relevance? Is number of occurrences of a word or the semantic distance between desired and available terms appropriate?

Reasons for the inaccuracy of searches include:

- The lack of common semantics between multiple sites, providers of objects, and users of objects

- Inexact classification, if any, of objects using the semantics that do exist

- Lack of reliable methods to compute relevance of objects that only satisfy some of the attributes that a user is looking for

The purpose of taxonomies is to develop a shared understanding of lists of terms, and relationships between these terms, encountered in work in some domain. Once there is a taxonomy, the search engine can make use of the shared vocabulary as well as exploit the relationships between terms in order to create meaningful relevance scores. A taxonomy may be multi-dimensional, which means that each object must be described using more than one collections of class terms.

A taxonomy-based organization and search system would facilitate working with taxonomies, including creating and maintaining the classification terms, classifying objects, and conducting searches. It would include a taxonomy browser, taxon-

3

omy editor, taxonomy-based search engine, an editor for search criteria specification, search algorithm, search algorithm parameters editor, and a results visualizer.

However, taxonomies—and the accompanying relationships—can be fairly complex. For example, in multi-level taxonomies, two terms may be related but it may require several steps to make the deduction that they are. In multi-dimensional taxonomies, multiple independent collections of terms may be required to classify the objects, each collection comprising a taxonomy. Developing the various components of a taxonomic system gets more complex when the taxonomy is dynamic, multi-variable, and multi-dimensional; it becomes particularly challenging in the context of developing Web-based taxonomic systems.

In this thesis, we give a formal logic-based specification of a taxonomy-based organization and search service. We focus on a particular kind of taxonomic system, where the taxonomy is restricted to a hierarchical arrangement of terms (with single-inheritance) but one which may consist of multiple independent taxonomies. The multi-taxonomy aspect of such systems complicates the design of its different components, including the browser, editor and the search and retrieval methods. Multiple inheritance would make things further complex, but we think our approach would generalize and scale up to that scenario as well.

Our interest in this area grew out of work on designing DecisionNet, a digital library of decision technologies. DecisionNet, as described in Section A, is currently implemented through common gateway interface (CGI) scripts. This implementation limits it to a single taxonometric dimension. The introduction of Java applets allows users to view and utilize the entire four level taxonomy.[Ref. 8]

D.   OUTLINE FOR REMAINDER OF THESIS

Chapter II discusses the taxonomy and defines relationships within it. Chapter III explores the challenges of searching and retrieving multi-variable, multi-dimensional objects. Chapter IV provides the analysis and design of the goal ar-

chitecture. Chapter V represents the actual implementation and the rationale for deviations from the goal architecture. Chapter VI is a tour of the system. Finally, Chapter VII identifies the difficulties and challenges that were overcome and those that remain.

II. TAXONOMY

A. TAXONOMY: FORMAL SPECIFICATION

Definition 1 (Taxonomy) *A taxonomy is a collection Γ of class terms used in classifying or describing a library of objects Λ in some domain. The classification is achieved via instances $\psi(\Gamma, \Lambda)$ of mapping relationships Ψ. Additionally, there may be instances $\omega(\Gamma, \Gamma)$ of structural relationships Ω involving the elements of Γ, and a collection of logical axioms Φ constraining these relationships. To be precise, each element in Ψ is a relationship that maps elements of Λ onto elements of Γ, and each element in Ω is relationship from the set Γ onto itself.*

Formally, we write a *taxonomy scheme* as a tuple

$$< \Omega, \Phi, \Psi > \tag{II.1}$$

and a *taxonomy instance* as a tuple

$$< \Gamma, \omega(\Gamma, \Gamma), \Phi, \Lambda, \psi(\Gamma, \Lambda) > \tag{II.2}$$

In this thesis, as mentioned above, our interest is in a special kind of taxonomic system. This system is defined by specifying the structure of Ω, Φ and Ψ. For the purpose of specifying the denotational semantics of the relationships, let Δ denote the universe of discourse (i.e., each object in Λ maps to an element of Δ) corresponding to the object collection Λ that needs to be classified and searched. Further, for any class term γ ($\in \Gamma$), let Δ_γ denote the objects (subset of Δ) belonging to class γ.

1. Relationships

We include the following relationships in our taxonomic system.

a. Ω

Taxonomy specification involves two relationships: **refines** and **syn**.

- **refines**: a binary relationship of the form γ_1 refines γ_2, where $\gamma_1, \gamma_2 \in \Gamma$. The informal interpretation is that all objects in the class γ_1 also belong to class γ_2. Formally, we get

$$\forall \gamma_1, \gamma_2 \quad (\gamma_1 \text{ refines } \gamma_2) \to (\Delta_{\gamma_1} \subseteq \Delta_{\gamma_2}) \tag{II.3}$$

- **syn**: a binary relationship of the form $\text{syn}(\gamma_1, \gamma_2)$, where $\gamma_1, \gamma_2 \in \Gamma$. The informal interpretation is that classes γ_1 and γ_2 are synonyms. Formally, the semantics are given as:

$$(\text{syn}(\gamma_1, \gamma_2)) \to (\Delta_{\gamma_1} = \Delta_{\gamma_2}) \tag{II.4}$$

Sometimes, synonyms may be used to force a hierarchical representation for a collection of terms and relationships that would otherwise require a network (multiple parents allowed). This is achieved by making a copy—synonym—of an object at multiple branches in the taxonomy tree.

In addition, we make use of a relationship **isA** (of the form γ_1 isA γ_2) which is derived from the relationship refines, except that isA is transitive (a complete definition is given via axioms II.8 and II.9). This relationship, strictly, is not required, but is included as syntactic sugar and for ease of exposition.

Finally, note that we can define a relationship **parent** which is the inverse of the isA (or **child**) relationship, and relationships **ancestor** and **descendant** that are transitive relationships based on the parent and child relationships, respectively.

b. Ψ

Object classification involves just one relationship **ins-of**.

- **ins-of**: a binary relationship of the form λ ins-of γ, where $\lambda \in \Lambda$ and $\gamma \in \Gamma$. The interpretation is that object λ in the domain belongs to the class γ in the taxonomy. Formally, we say

$$(\lambda \text{ ins-of } \gamma) \to (\lambda \in \Delta_\gamma) \tag{II.5}$$

As above, we also allow a binary relationship **sib** (of the form $\text{sib}(\lambda_1, \lambda_2)$, derived from the relationship ins-of, to indicate that λ_1 and λ_2 are siblings (have the same parent); Axiom II.14 further clarifies the meaning of this relationship. This relationship, strictly, is also not required, but is included for clarity.

8

2. Axioms, Φ

The formal semantics of, and integrity constraints on, the relationships stated above can be stated clearly via the following axioms.

refines: subclass hierarchy, single-parent hierarchy All objects that are an instance of a certain class are also instances of its parent class. Second, each term in Γ can be a refinement of at most one (parent) term. The root term has no parent.

$$((\lambda \text{ ins-of } \gamma_1) \wedge (\gamma_1 \text{ refines } \gamma_2)) \rightarrow (\lambda \text{ ins-of } \gamma_2) \qquad (II.6)$$

$$((\gamma_1 \text{ refines } \gamma_2) \wedge (\gamma_1 \text{ refines } \gamma_3)) \rightarrow (\gamma_2 = \gamma_3) \qquad (II.7)$$

isA: refines, transitive

$$(\gamma_1 \text{ refines } \gamma_2) \rightarrow (\gamma_1 \text{ isA } \gamma_2) \qquad (II.8)$$

$$((\gamma_1 \text{ isA } \gamma_2) \wedge (\gamma_2 \text{ isA } \gamma_3)) \rightarrow (\gamma_1 \text{ isA } \gamma_3) \qquad (II.9)$$

In terms of the universe of discourse,

$$(\gamma_1 \text{ isA } \gamma_2) \leftrightarrow (\Delta_{\gamma_1} \subseteq \Delta_{\gamma_2}) \qquad (II.10)$$

synonym: equivalence relation

$$\text{syn}(\gamma_1, \gamma_1) \qquad (II.11)$$

$$\text{syn}(\gamma_1, \gamma_2) \rightarrow \text{syn}(\gamma_2, \gamma_1) \qquad (II.12)$$

$$(\text{syn}(\gamma_1, \gamma_2) \wedge \text{syn}(\gamma_2, \gamma_3)) \rightarrow \text{syn}(\gamma_1, \gamma_3) \qquad (II.13)$$

sibling

$$((\lambda_1 \text{ ins-of } \gamma) \wedge (\lambda_2 \text{ ins-of } \gamma)) \leftrightarrow \text{sib}(\lambda_1, \lambda_2) \qquad (II.14)$$

3. Multi-Dimensional Taxonomic System

A multi-dimensional taxonomic classification system is one in which a collection of objects Λ must be classified along more than one dimension, and where there is an independent taxonomy on each dimension. In an n-dimensional taxonomic system, let us denote the collection of terms in the taxonomies by $\Gamma^1 \ldots \Gamma^n$. If we assume that all dimensions are governed by the same relationship schemes (Ω, Φ) and axioms (Ψ), then a multi-dimensional taxonomic scheme can be written as the tuple

9

$$< \Omega, \Phi, \Psi > \qquad \text{(II.15)}$$

and a taxonomy instance as a tuple

$$< (\Gamma^1, \ldots, \Gamma^n), (\omega(\Gamma^1, \Gamma^1), \ldots, \omega(\Gamma^n, \Gamma^n)), \Phi, \Lambda, (\psi(\Gamma^1, \Lambda), \ldots, \psi(\Gamma^n, \Lambda) > \qquad \text{(II.16)}$$

which may, alternately, be written as a collection of tuples:

$$< \Gamma^1, \omega(\Gamma^1, \Gamma^1), \Phi, \Lambda, \psi(\Gamma^1, \Lambda) >, \ldots, < \Gamma^n, \omega(\Gamma^n, \Gamma^n), \Phi, \Lambda, \psi(\Gamma^n, \Lambda) > \qquad \text{(II.17)}$$

B. TAXONOMY DEVELOPMENT AND OBJECT CLASSIFICATION

1. Taxonomy Development

For a single taxonomy, this means specifying

- elements of Γ

- instances of the relationships in Ω, (i.e., instances of refines and syn).

For example, consider the "problem type" taxonomy tree from the GAMS system, [Ref. 9] displayed in Figure 1. Elements of Γ are all the terms appearing in the list. The subclass relationships are represented using the refines relationship, e.g. (using the shorthand codes from the taxonomy),

```
refines(G1,G)
     . . .
refines(G2,G)
refines(G2a,G2)
refines(G2b,G2)
refines(G2b,G2)
refines(G2c1,G2c)
refines(G2c2,G2c)
     . . .
```

```
[F] Solution of nonlinear equations
[G] Optimization (search also classes K, L8)
    [G1] Unconstrained
    [G2] Constrained
        [G2a] Linear programming
        [G2b] Transportation and assignments problem
        [G2c] Integer programming
           [G2c1] Zero/one
          [G2c2] Covering and packing problems
           [G2c3] Knapsack problems
          [G2c4] Matching problems
           [G2c5] Routing, scheduling, location problems
                [G2c6] Pure integer programming
                 [G2c7] Mixed integer programming
          [G2d] Network (for network reliability search class M)
          [G2e] Quadratic programming
          [G2f] Geometric programming
          [G2g] Dynamic programming
         [G2h] General nonlinear programming
        [G2i] Global solution to nonconvex problems
      [G3] Optimal control
      [G4] Service routine
[H] Differentiation, integration
[I] Differential and integral equations
[J] Integral transforms
[K] Approximation (search also class L8)
```

Figure 1. Fragment of "problem type" taxonomy from GAMS system

For an n-dimensional taxonomy, this process must be repeated for each of the n dimensions.

Note also that there is an obvious database implementation of a taxonomy in terms of these two relations.

2.   Object Classification

Classification of objects $\lambda \in \Lambda$ involves specifying instances of the relationships in Ψ, i.e., one instance of the ins-of relationship for each $\lambda \in \Lambda$.

11

Modules for class G2c1
Package NAG at GRANTA

- H02BBE Solves zero-one, general, mixed, or all (pure) integer programming problems using ...

- H02BBF Solves zero-one, general, mixed, or all (pure) integer programming problems using ...

Figure 2. Fragment of object classification in GAMS taxonomy

which would be represented as

```
ins-of(G2c1,H02BBE)
ins-of(G2c1,H02BBF)
...
```

III. SEARCH AND RETRIEVAL IN A TAXONOMIC SYSTEM

Assume that an object receives a score of 1 if it matches exactly the search criteria specified by the user, and a score of 0 when it has no relevance to the user's requirements.

A. SEARCH CRITERIA SPECIFICATION

Let κ denote the criteria for searching and determining relevance of objects in a taxonomy. For a single-dimensional taxonomy, κ simply specifies a class $\widehat{\gamma} \in \Gamma$. If we denote the dimension by δ, then κ may be written as

$$\delta = \widehat{\gamma} \tag{III.1}$$

which specifies a search for objects that belong to, or otherwise are relevant to, class $\widehat{\gamma}$. For a multi-dimensional taxonomy with dimensions $\delta_1, \ldots, \delta_n$, we write the criteria κ_i for dimension i as

$$(\delta_i = \widehat{\gamma_i}) \tag{III.2}$$

whereas the complete search criteria κ is specified as

$$(\delta_1 = \widehat{\gamma_1}) \wedge \ldots \wedge (\delta_n = \widehat{\gamma_n}) \tag{III.3}$$

B. RELEVANCE FUNCTION

We can state, without loss of generality, that if an object perfectly matches a search criteria κ, it receives a score of 1, and that if it matches perfectly the criteria κ_i for dimension i then it receives a score of 1 on that dimension. Given a search criteria, the overall relevance score θ of an object is computed as a weighted function—whose

13

values fall in the [0,1] interval—of its relevance scores on each dimension. The general form of such a function can be given as:

$$\theta(\lambda|\kappa) = \sum_i w_i \cdot \theta_i(\lambda|\kappa_i) \qquad \text{(III.4)}$$

where θ_i is the score on dimension i and where the weights w_i sum to 1. Note that instead of $\sum$ one could use some other aggregation operator.

To complete the definition of a relevance function, we need to define how to compute the relevance score on each dimension. Many such functions are possible, and in general we can assume that they work by measuring how the object is related to the desired class, and how close the relation is.

- Relevance: In our taxonomic system, an object that does not match the search criteria exactly (i.e., it is not an instance of the desired class) may still be relevant because it belongs to a *parent*, *child* or *sibling* class of the ideal class. We stated these relationships, all of which draw from the base relation *refines*, in Chapter II Section A.

  Of course, being a parent may not be of equal relevance as being a child or a sibling. In each case, the relevance score must be suitably degraded. Therefore, we define separate (0,1) parameters σ_p, σ_c and σ_s as the degradation multipliers for objects belonging to the parent, child and sibling, classes of the desired class.

- Distance: Objects that belong to a *parent*, *child* or *sibling* class of the ideal class are, obviously, only one level from the ideal. More generally, objects that belong to *ancestor* and *descendant* classes (and we do not consider *cousins*) are m levels removed from the ideal class, and their relevance should decrease as the distance increases.

  Here, we define a partial function m_{ref} to compute the levels between two classes. (Note that the function is undefined when the two classes are not on the same root path.)

$$
\begin{aligned}
\gamma_1 \text{ refines } \gamma_2 \quad &\rightarrow \quad m_{ref}(\gamma_1, \gamma_2) = 1 & \text{(III.5)} \\
m_{ref}(\gamma, \gamma) \quad &= \quad 0 & \text{(III.6)} \\
m_{ref}(\gamma_1, \gamma_2) \quad &= \quad -m_{ref}(\gamma_2, \gamma_1) & \text{(III.7)} \\
m_{ref}(\gamma_1, \gamma) + m_{ref}(\gamma, \gamma_2) \quad &= \quad m_{ref}(\gamma_1, \gamma_2) & \text{(III.8)} \\
& & \text{(III.9)}
\end{aligned}
$$

14

The relevance score θ_i on dimension i, given search class $\widehat{\gamma}_i$, of an object λ which belongs to class γ is

$$\theta_i(\lambda|\kappa_i) = \left\{ \begin{array}{ll} \sigma_c^m & \text{if } m_{ref}(\gamma, \widehat{\gamma}_i \geq 0 \\ \sigma_p^m & \text{if } m_{ref}(\gamma, \widehat{\gamma}_i \leq 0 \\ \sigma_s & \text{if } \gamma \text{ is a sibling of } \gamma_i \\ 0 & \text{otherwise} \end{array} \right\} \qquad \text{(III.10)}$$

Recall that the overall relevance score θ is computed by combining the scores θ_i as in Equation III.4.

It may now be obvious that the relevance function θ resembles a probability function—the relevance score $\theta(\lambda|\kappa)$ can be interpreted as the probability that an object λ would be relevant to a user who specified the search criteria κ. The function can easily be customized by users via the following input parameters:

- Weights $w_1, \ldots, w_n$ for dimensions

- Degradation multipliers σ_c, σ_p, and σ_s

C. SEARCH ALGORITHM: EFFICIENCY AND COMPLEXITY

The objective of a search algorithm is to find the x most relevant objects in an object library, based on a variety of factors listed below, but where x itself is a parameter that cannot be specified in advance with any degree of credibility. The algorithm must employ methods to prune and terminate the search (worst case is to compute a relevance score for every object in the library, but this is not desirable). For example, one may search only up to two levels from the desired class. Or, perhaps, dynamically determine the search depth in a dimension as some function of the weight of that dimension. These search strategies may be specified as meta-rules in the search system, and can make the search much more efficient.

15

D. COMPONENTS OF A TAXONOMY-BASED OBJECT CLASSIFICATION AND RETRIEVAL SYSTEM

We have developed the mathematical and taxonometric framework to solve this problem. We will now translate that framework into components that are required to perform the various functions. The following list describes the components in broad terms. Chapter V will map these components to those developed for this thesis.

- Taxonomy Browser

  This module allows users to navigate a single or multi-dimensional taxonomy. A hierarchical taxonomy can be displayed in many ways, including as an indented list (the approach employed in GAMS), as a tree or hierarchy, and as a cascading list (the approach employed in this thesis).

- Object Classification Editor

  This module allows users (object providers) to classify newly entered objects, or to change classification of existing objects, in terms of the classes in the taxonomy. This usually requires a taxonomy browser to allow users to determine which classes in the taxonomy are most suitable for the object to be classified.

- Search Criteria Editor

  This module allows users (consumers) to specify the search criteria to be used in retrieving objects from the library. The criteria are limited to terms contained in the taxonomy.

- Search Algorithm

  This algorithm retrieves a list of objects from the library, based on the relevance scores of objects, computed on the basis of the search criteria specified by the user and the relationships in the taxonomy and object databases. For objects that do not exactly meet the user's criteria, the relevance is computed based on the degradation parameters for inexact matches.

- Search Parameters Editor

  This module lets the users specify the weights for the different attributes, as well as the values of the degradation parameters used in the relevance function in the search algorithm.

- Search Results Visualizer

  This module displays the results of the search function. Options include different ways of sorting the results, recomputing relevance scores if user changes some of the parameter values, and displaying the results in different ways.

IV.   ANALYSIS AND DESIGN

Recall that the objectives of this thesis are the classification, registration, search, and retrieval of multi-variable digital objects. The design shown here achieves these goals within the limitations that will be discussed. The assumptions and decisions that were made during the analysis and design process are discussed here.

A.   ASSUMPTIONS

A number of technologies are available today to develop a WWW application. Performance of the system depends both on the development technology and the technical capabilities of the user (e.g., user's Web browser and type of Internet connection). In developing our system, we made certain assumptions about user capabilities before selecting the development technologies. The user is assumed to have a Java capable Web browser. In most cases this translates to either Internet Explorer or Netscape Navigator (versions 2.X and higher). To access the system the user must have some knowledge of the WWW. Prior knowledge of Decision Support Systems (DSS) is not required. Access is assumed to be through a 28.8 kb/s modem.

B.   DESIGN CRITERIA

The design criteria are also goals set for this system. The design criteria are shown in the following list.

- Platform Independence
- Usability
- Generalization

Platform independence means that anyone with a Java capable Web browser and Internet access will be able to use this system. The hardware, operating system, and Web browser are irrelevant. The Java Virtual Machine (JVM) will handle the

17

differences that have made *platform independence* impossible in the past. Java, with its promise of "Write Once, Run Anywhere", is the primary programming language. [Ref. 10] The system must be accessible to the average user to satisfy the second design criteria.

Usability, as we use it in this thesis, is defined by the user. The graphical user interface (GUI) is intuitive and easy to navigate. The system runs over the WWW and does not require any new software to be installed on the client machine. Efficiency and speed are acceptable to the average user. This factors must be met or people will not use the system.

Generalization allows the system to display any taxonomy that conforms to the size and depth limits. The DecisionNet project calls for a five variable, four dimension system. All five variables do not extend to four dimensions in this case. New taxonomies may have fewer than five variables and fewer than four dimensions. The only limit imposed is the upper bound. Beyond the taxonomy, *generalization* also means that new CGI scripts can be called to achieve the registration and search functions without altering the Java applets. Input parameters to the applets allow this flexibility. The input parameters are discussed in Appendix C.

C.   FUNCTIONALITY

There are four main functions performed by the system. They are:

- Classification
- Registration
- Search
- Retrieval

Classification of objects is a function performed by producers of decision technologies. *Classification* allows the producer to define his technology within the DecisionNet taxonomy. The accurate placement within the taxonomy is essential prior

18

to *registration*. The applet's multi-level display capability allows the use of the full taxonomy that was not previously possible. The taxonomy allows the *classification* of an object (DSS in this case) in each of five attribute areas. Each attribute may extend as many as four levels down in a hierarchical structure that is defined by parent/child relationships. The *registration* function takes the results of the *classification* and enters them in the technology object database.

The current system is capable of search by one variable only, with binary or yes/no results. An algorithm to examine all of the possible relationships is needed to produce a relevancy score. This would give a relative distance from the search criteria for objects of possible value to the user. A user also requires the ability to define a weight for each attribute as well as degradation multipliers for relationships other than an exact match. These relationships are defined in Chapter II Section A.

The last functional area is retrieval. This could also have been labeled display. The search results must be returned in a format that is easily interpreted and usable.

D.   CLIENT VS. SERVER

In a client/server environment there are decisions to be made about where best to locate components of a system. Each node has pros and cons that influence design decisions. A node, in this context, is simply a computer. A node can be a client, a server, or both. With client machines getting more capability every day there is a tendency to push components out to the client whenever possible.

This client centric approach works well when bandwidth is not a concern. The client/server options must be consistent with the future expansion of DecisionNet. The system must be scaleable and robust. The assumption that users access DecisionNet through a modem requires that download time be a consideration. Server load and capabilities are another driving factor. The main DecisionNet server is a Pentium (90 MHz) with 24 MB of random access memory (RAM). The current hardware is adequate, but it can quickly be bogged down with several simultaneous users.

Option A (Current system) - Server Intensive
HTML / CGI Scripts

Dnet
DB1 & DB2
Search Algorithm
(server)

User
(client)

Request

Taxonomy (html forms)

Search Criteria

Answer

Write to DB for Registration

DB1=Taxonomy DB
DB2=Technology Object DB

Considerations:
-server intensive
-low user interactivity
-no browsing of the taxonomy
-no relevancy scores of searches
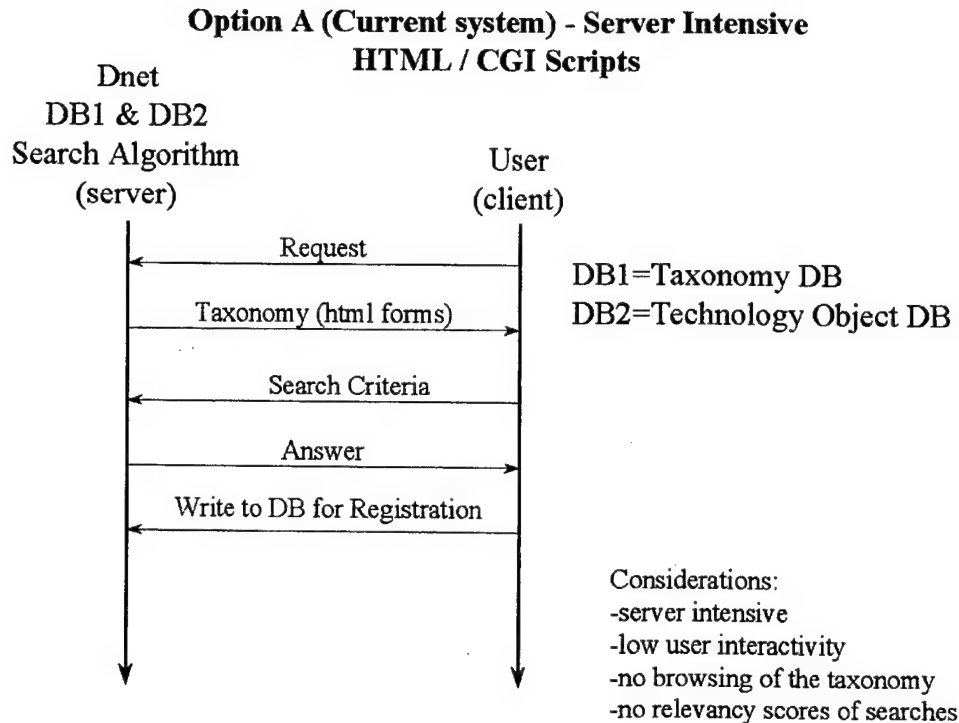
Figure 3. Time Sequence Diagram - Option A

With these factors in mind we examine three options.

1.   Option A - Server Intensive

The first option is a description of many current CGI systems. The processing load is almost entirely on the server node. The client browser does little more than display hypertext mark-up language (HTML) pages. The server dynamically generates HTML pages through CGI scripts. Figure 3 describes the interaction and the location of various components. There is no significant processing on the client node. The CGI scripts generate content dynamically, but it is static once it reaches the client. This produces very low interactivity.

2.   Option B - Client Intensive

In direct contrast to Option A, Option B is almost entirely on the client node. The server is little more than a file and database server. In a networked environment with 10 MB/s - 1 GB/s data transfer rates this option would be a good one. With

Option B - Client Intensive
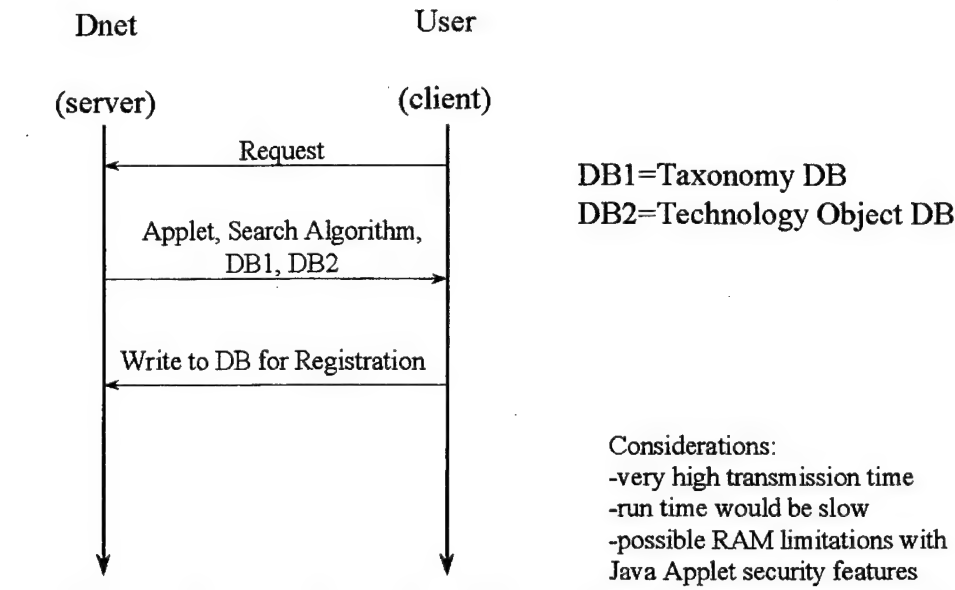
Figure 4. Time Sequence Diagram - Option B

users accessing the system with 28.8 kb/s modems it will prove to be unusable due to the long transmission times. Figure 4 describes the interaction and the location of various components.

As both the taxonomy and object databases grow, the situation will get worse. The security features of Java applets could become a limiting factor. By design, applets are downloaded, stored, and run entirely in the client's RAM. A security feature of applets is their inability to read or write from any drive except the server from which they were launched. With large databases and applet code, a client may not be able use the system due to memory limitations. The ability to access a client's local drive is being considered for a future Java release.

3.   Option C - Goal Architecture

Option C combines the benefits of Options A and B. The components whose functions do not rely on repeated server communications, and can be economically

21
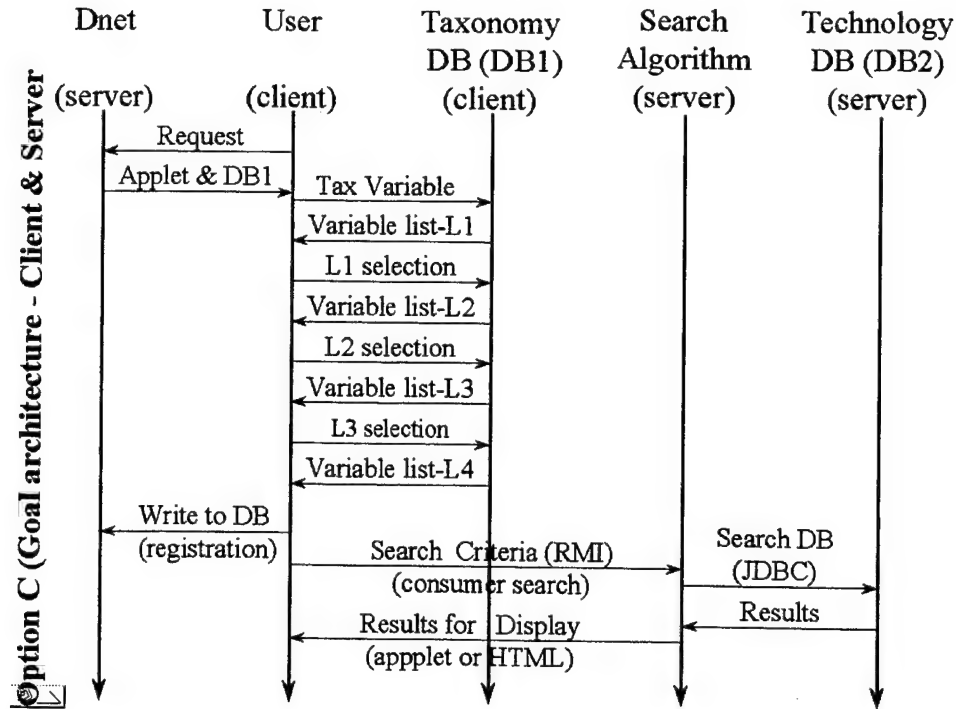
Figure 5. Time Sequence Diagram - Option C

transmitted, are located on the client node. Functions that fail either of those criteria are located on the server node. This approach is a three tiered client/server architecture. [Ref. 11] The taxonomy database and the applet code reside on one server. The technology object database and search algorithm reside on a second server. Communication is limited once the applet is loaded onto the client. Figure 5 describes the interaction and the location of various components.

There are several factors that make the three tiered approach attractive. The taxonomy database is closely tied to the applet. It is also relatively small and stable. It is much more efficient to send the taxonomy with the applet that will use it than to have repeated calls back to the server. This approach allows repeated browsing with no additional overhead.

The search algorithm is closely tied to the technology object database. This database could become quite large resulting in transmission delays if it were sent to the client. The search algorithm could also be a bottle-neck. The search algorithm

22
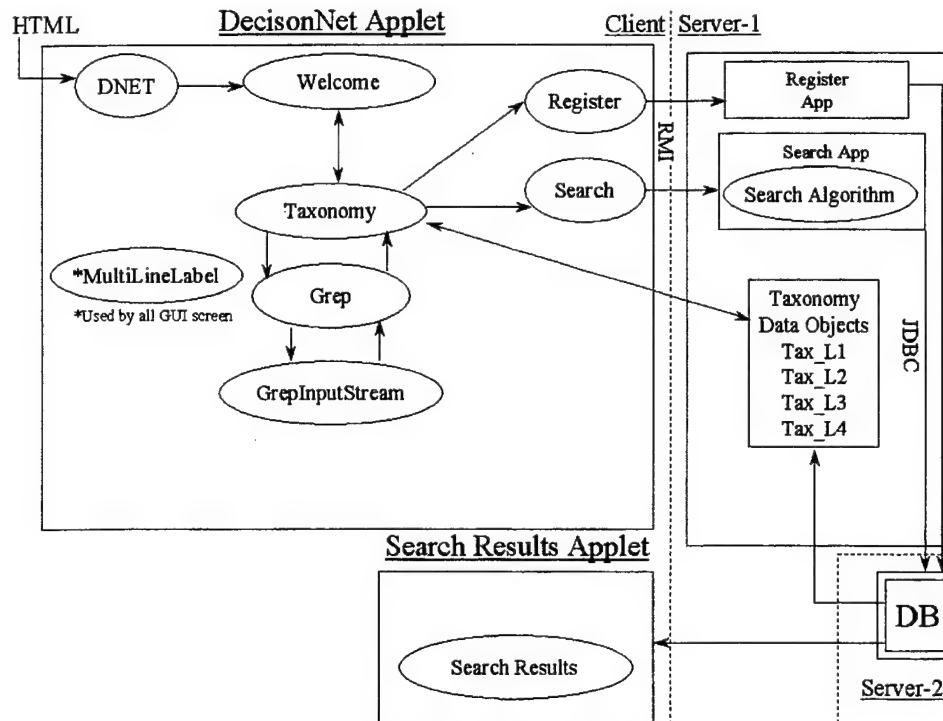
Figure 6. Goal Architecture

is computationally intensive and makes repeated queries to several databases. These facts, combined with the transmission time, could make the search performance unacceptable.

E.   THE ALL-JAVA SOLUTION

Sun's Java Developer's Kit (JDK) 1.1.2 supports a three tiered client/ server architecture. The additions of Remote Method Invocation (RMI) and Java DataBase Connectivity (JDBC) make the three tiered design possible. The applet security restrictions are enforced by only allowing the applet to communicate with the server that launches it. The applet server makes calls to the database server on the applet's behalf. The applet is limited in what it can do because the methods available to it are defined in the interface running on the applet server. The applet server acts as an intermediary between the applet and the object database server.

This design meets the technical goals that we established and took advantage of

the strengths of both client and server nodes. However, the goal architecture currently fails the test of usability. The only browser that currently supports JDK 1.1.2 is Sun's HotJava browser. Netscape Communicator and Microsoft Internet Explorer dominate the market of Web browsers. Neither one has fully implemented the latest Java release and neither currently supports this design. Chapter V describes the actual implementation and discusses areas where it differs from the ideal technical solution shown here.

V.   IMPLEMENTATION

As discussed in Chapter IV, the goal of an all-Java, three tiered client server solution is not currently feasible. Without Web browser support, the design would not meet user needs, and thus fail the second design criteria of usability. The decision was made to modify the design to include a CGI based search algorithm on the database server. A suitable existing algorithm could not be located. The existing DecisionNet system consists of CGI scripts written in Object Pascal using the Delphi development tool. To maintain consistency with the existing system the search engine is written in Delphi.

Figure 7 illustrates the component location and system interaction. The only non-Java component is the search algorithm which will be discussed further in Section C.

Java Integration with existing Dnet Components

1 - Object Library Browser (future project-java applet)
2 - Taxonomy Browser (java applet)
3 - Object Classification Editor (utilizes #2)
4- Registration Object (existing Delphi CGI script-uses #3)
5 - Search Criteria Editor (utilizes #2)
6 - Search Algorithm (delphi CGI script)
7 - Search Parameters Editor (java applet)
8 - Search Results Visualizer (java applet)

Figure 7. System Integration

25

A. TAXONOMY DATA FILES

The original design called for the taxonomy to be read dynamically from the database. The relatively static nature of the taxonomy allows the information to be stored in data files. If the applet were to query the database it would have to be launched from the same server to satisfy applet security restrictions. We felt that there was no advantage to adding this restriction. Updating the taxonomy involves nothing more than editing a text file. By not accessing the database, the download time is greatly reduced. We divide the four levels of the taxonomy into separate data files. The first level is read in and displayed while the others are loading instead of waiting for the entire taxonomy before display.

The data files are text files with a standard format. The parent term starts at character position 1 and the child term starts at character position 39. This format is required by the applet for correct parsing of the data. This format accommodates the longest string in the taxonomy in addition to a small buffer. This design feature allows any taxonomy (within design specifications) to be displayed with this system.

B. JAVA CLASSES

Java source code has a .java extension. Compiled Java code has a .class extension. A class file is analogous to an executable file in other languages although it can not run without the JVM. Java's platform independence is achieved by compiling source code to an intermediate level known as byte codes. The byte codes are object code for the JVM. The JVM interprets the object code to execute on the local platform.

The following is a description of each class' function and its interaction with the overall system. An object is an instance of a Java class. The distinction is small and they are often used interchangeably. The term object can also be used as a generic label for a decision technology in this system. We refer to attributes as instances of variables in much the same way. A variable is a generic term such as 'Problem Area'.

An attribute is a child of that variable describing a particular technology object. Where appropriate, the *DecisionNet Java Objects* refer to the general *Components* described in Chapter III Section D.

1.   Dnet_Reg.class

This applet allows a producer to explore the taxonomy and decide where his decision technology fits into it. This object implements the *Object Classification Editor.* (See Chapter III Section D) Once the technology is correctly classified, the applet interfaces with an existing CGI script (regteca.exe) [Ref. 7] to complete the registration. The applet allows specification down to four levels for each variable. The applet captures user information from the HTML page that launches it. The user information is required by the CGI script (regteca.exe). It utilizes the *taxonomy object* as its main GUI.

2.   Dnet_Search.class

This applet allows a consumer to explore the taxonomy and specify search criteria for the five variables. This object serves as the *Search Criteria Editor.* (See Chapter III Section D) The applet allows specification down to four levels for each variable. The applet utilizes the *taxonomy object* as its main GUI. It passes control and the search criteria to the *weights object.*

3.   Dnet_Browse.class

This applet allows a consumer to browse the taxonomy . This object serves as the *Taxonomy Browser.* (See Chapter III Section D) The applet allows browsing down to four levels for each variable. The applet utilizes the *taxonomy object* as its main GUI.

4.   Taxonomy.class

The taxonomy object is utilized by all three applets. It is the main GUI that displays the taxonomy and highlights user selections. The user controls the display by double clicking the left mouse button within a displayed list. If there are any

27

children of the selected term in the taxonomy, they are displayed in the adjacent list. This cascading refinement is highlighted in the text boxes above the lists.

Appendix C contains a list of variables that are passed to the *taxonomy object* when one of the three applets calls it. Any parameters that do not apply are sent as a single space. All of these parameters originate in the HTML page that launches the applet. The applet reads them and passes the appropriate parameters to the other objects (taxonomy, weights, grep, etc.) as needed.

5.   Weights.class

The *weights object* accepts the search criteria and allows the user to adjust the variable weights and degradation multipliers used by the search algorithm. This object serves as the *Search Parameters Editor*. (See Chapter III Section D) Default values are provided through the input parameters. Equal weight is given to each variable and degradation multipliers are set to the expected probable value of a given relationship. The *weights object* passes control to the *Search Algorithm* (Indexed.exe) described in Section C.

6.   Search_Results.class

The *Search_Results object* accepts the output from the *Indexed.exe* script for display within the applet. This object serves as the *Search Results Visualizer*. (See Chapter III Section D) The datastream is parsed and formatted using control characters placed in the stream by the *Indexed.exe* script. The user's search criteria and variable weights are returned for review. Each relevant technology is displayed with a total score as well as individual attribute scores. This allows the user to see which attributes produced a relationship, and what type of relationship it is.

7.   Reg_Complete.class

The *Reg_Complete object* is a simple dialog screen that displays the success or failure of various operations. It is utilized by the *DNET_Reg* and *DNET_Search* applets.

28

8. MultiLineLabel.class

At the time of coding, the Java language did not define a label of more than one line of text. This object provides that functionality. [Ref. 10]

9. Grep.class

This object is slightly modified from the cited source. Its only function is to discard lines that do not contained a given string. It receives lines of text from the *GrepInputStream object* along with the string to be matched. [Ref. 10]

10. GrepInputStream.class

This object is slightly modified from the cited source. Its only function is to read lines from an input stream. It passes each line to the *Grep object* along with the string to be matched. [Ref. 10]

11. QueryString.class

This object concatenates the variables into one string to be sent to either the registration (*regteca.exe*) or search (*Indexed.exe*) CGI script.

C. DELPHI SEARCH SCRIPT

The only component of this system that is not implemented in Java is the search algorithm. Once the decision to modify the all-Java solution was made, a Java search algorithm did not offer any advantages. The driving factor in going to a native code solution was speed. The search algorithm is computationally intensive and makes repeated queries to several database tables. The performance would be unacceptable given the current speed of Java code. Java code can be significantly slower than C code on average. [Ref. 10] Sun is currently working to improve the performance with the introduction of a Just-in-Time (JIT) Compiler.

1. Indexed.exe

This is the implementation of the *Search Algorithm*. (See Chapter III Section D) The search algorithm is designed to compute a relevancy score for technolo-

gies that may be of interest to a consumer conducting a search. The goal is to limit the number of records that need to be processed without discarding any that may be useful. The initial screening looks to find at least one direct match between the search criteria and a record attribute. Those records with a match are then eligible to be scored. As the number of records in the Technolo.db [Ref. 7] database increases this screening can be increased to require two or more matches before evaluating the record's attribute relationships.

For every record that is eligible, the algorithm checks each attribute to determine if a relationship exists. The order of precedence follows:

- Search Criteria = ALL, indicating no preference
- Search Criteria is a direct match of the Attribute
- Search Criteria is a child of the Attribute
- Search Criteria is a parent of the Attribute
- Search Criteria is a sibling of the Attribute
- Search Criteria is a descendant of the Attribute
- Search Criteria is a ancestor of the Attribute

Queries are conducted through boolean gates that stop the process once a relationship has been established. Every eligible record is assigned five attribute scores. Once all of the relationships are established the algorithm computes a total score for each record. The relevancy score is a summation of the variable weights multiplied by the attribute score. This is shown notationally in Equation III.4.

This implementation takes advantage of the strengths of each node of the client/server environment. The components are closely and logically tied to their data sources. Integration with the current system drove the continued use of CGI scripts. For most functions CGI scripts are efficient and robust, but they lack interaction. The combination of Java Applets and CGI scripts offers a reasonable alternative.

VI.   SYSTEM TOUR

The classification, registration, search, and retrieval functions are visual and intuitive. Once the applet is launched, a mouse is generally all that is required to navigate among the functions. The only time that the keyboard is required is to change the default values in the search engine. The following sections show the flow of control through the system as well as the various screens that a user will see.

A.   TECHNOLOGY REGISTRATION

Producers of decision support technologies use this applet to correctly classify their decision technology within the DecisionNet taxonomy. (See Figure 8) All of the user information is supplied to the applet by the DecisionNet CGI script that produced the HTML page. Once the object classification is complete, the user presses the 'Submit for Registration' button and control passes to the registration CGI script (regteca.exe). The success or failure of the registration process is displayed in Figure 9.

B.   CONSUMER SEARCH

This applet allows a consumer to conduct a detailed search of the technology object database. The search engine allows the user to be as vague or as specific as he would like. He may specify any or all of the attributes that he is interested in. The default value of 'ALL' will return a much wider range of responses than will a more specific selection.

The *Search Criteria Editor* utilizes the same GUI shown in Figure 8 for its initial display. In the search configuration, the labels, instructions, and button functions have changed to reflect this role. This GUI allows the user to view and select from the entire taxonomy. Once the terms have been selected, the consumer pressed the 'Define Search Weights' button. Control then passes to the *Search Parameters Editor* shown in Figure 10.

31

Figure 8. Object Classification Editor

The *Search Parameters Editor* allows a consumer to further refine their search. If certain variables are more or less important, the default weights should be changed. The five variable weights must sum to 1.0 or the results will be meaningless. The degradation factors allow a consumer to state the utility of various relationships that may exist between their search criteria and object attributes. The degradation multiplier is applied as follows:

- If the criteria is a child of the attribute then the attribute receives a score equal to the child degradation multiplier.

- If the criteria is a parent of the attribute then the attribute receives a score equal to the parent degradation multiplier.

- If the criteria is a sibling of the attribute then the attribute receives a score equal to the sibling degradation multiplier.

32
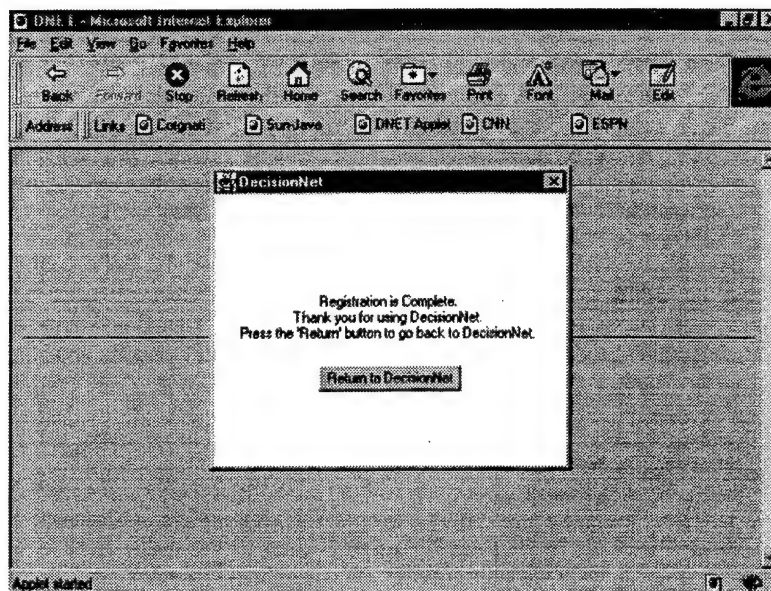
Figure 9. Results Dialog Window

- If the criteria is a descendant of the attribute then the attribute receives a score equal to the child degradation multiplier squared.

- If the criteria is a ancestor of the attribute then the attribute receives a score equal to the parent degradation multiplier squared.

Once the consumer has fully specified his search, he uses the 'Submit to Search Engine' button to launch the search. The search algorithm is run through a CGI script and produces an output file that is returned to the applet. The script ends each line with the control characters '/&'. This allows the applet to parse the stream of characters and recreate the form of the output file. The results are display in the *Search Results Visualizer* shown in Figure 11.

33

Figure 10. Search Parameters Editor

C.   TAXONOMY BROWSING

This is the simplest of the three applets. It is used to browse the entire five variable, four dimensional taxonomy. The user can explore the possibilities and depth available in the system. Once the browser applet is downloaded, there is no need for further communication with the server. It will run entirely on the client. Figure 8 again shows the taxonomy object. In the browser configuration, the labels, directions, and button functions have changed to reflect this role. The layout and control are identical.
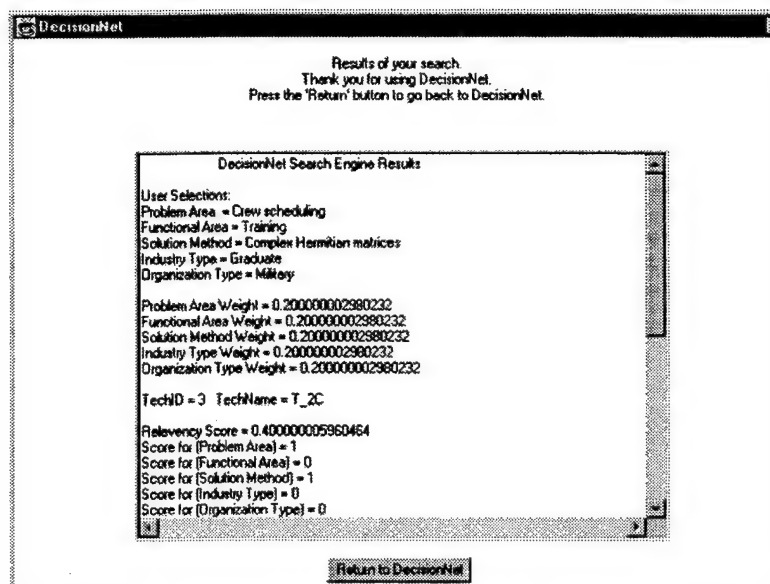
Figure 11. Search Results Visualizer

35

VII. CONCLUSIONS

A. CONTRIBUTIONS

This thesis defines a methodology for classifying, registering, searching, and retrieving multi-variable objects in a digital repository. The methodology includes the definition of relationships that exist in a parent/child taxonomy. The components required to implement such a system are described in general terms and are demonstrated within DecisionNet. This framework can be applied to any system that classifies objects with a taxonomy and stores them in a digital library.

The system developed for DecisionNet implements this methodology. It is both a testbed and a proof of concept. The ability to utilize the full taxonomy greatly enhances DecisionNet's ability to bring producers and consumers of decision technologies together. The generalization that is integral to this thesis enables it to be applied to any systems that relies on a taxonometric classification scheme. The number of attributes and levels can be increased in the future to accommodate a larger taxonomy.

B. LESSONS LEARNED

The following is a list of lessons learned along the thesis path.

- Repositories and code reuse hold a lot of promise for future projects. Java is not yet at this stage.

- The conceptual understanding of what object oriented programming is and what it offers does not come immediately. This basic understanding is required before any real progress can be made.

- Each decision made during analysis, design, and implementation effects the options available later in the development cycle. A good example is a naming convention. Planning and thought need to go into variable and object names before they become too cumbersome to change.

- Documentation should be written simultaneously with the code. This proved very beneficial when trying to integrate Java code with Delphi code.

- Java is not a mature language. Java is developing quickly, but it has not reached the maturity level of many object oriented languages. Java lacks a good IDE that allows for the drag and drop construction of GUIs. There are attempts being made, but they currently lag behind those developed for other languages.

- The key to running an applet within a web browser is the JVM. The JVM should interpret the applet bytecodes to run on any hardware and produce the same output regardless of the platform. This is not the case. There is a lack of standardization in the JVM implementation. This system was tested on several hardware platforms with different web browser combinations. The results were inconsistent. The basic functionality was adequate, but the display was awkward at times.

C. LOOKING FORWARD

This Thesis addresses and solves several areas that were identified in past DecisionNet thesis projects. The following topics are areas that remain for further research.

1. Search Algorithm

The search algorithm developed in this thesis establishes relationships for each attribute of a record that passes its initial screening. This works well in the current environment. As the object database grows in size, the efficiency of this approach will become a limiting factor. A clever indexing scheme will greatly improve the search efficiency. Another possibility is to establish relationships within the taxonomy and include them as an attribute in the taxonomy database. Chapter III Section C describes the dynamic determination of search depth based on search weight input. This will also improve the algorithm's efficiency.

2. Search Results Visualizer

The *Search Results Visualizer* is currently a display of the output text file from the search algorithm's CGI script. This data could be sorted and displayed in any number of ways that may be helpful to a consumer. The following is a list of possible representations that would greatly enhance the impact and usability of the returned

data.

- Display total scores first, and allow the user to drill down to review individual attribute scores of promising technologies

- A comparative bar chart to visually compare the relevance of different technologies against each other and against the ideal case of 1.0

- A pie chart to display the user defined variable weights

- A sliding scale that could be dragged with a mouse to perform sensitivity analysis on weights and degradation factors

An applet is the perfect vehicle to handle these tasks on the client machine.

3.   Three Tiered Client/Server

The implementation of the three tiered client server design will be possible in the next 12 - 24 months. As the Java language and the support for it grows, an all-Java solution will take full advantage of distributed computing and storage of information.

4.   Help Menus

The ability to right mouse click on a term or component would greatly add to the usability of the system. There are many technical and mathematical terms in the taxonomy that will not be familiar to the average user. Online amplification and definition would increase the effectiveness of the classification and search functions by answering questions as the arise.

APPENDIX A. SYMBOLS AND ACRONYMS

Symbol => Name

Symbol	Name
Δ	Delta
δ	delta
Γ	Gamma
γ	gamma
κ	kappa
Λ	Lambda
λ	lambda
Ω	Omega
ω	omega
Φ	Phi
Ψ	Psi
ψ	psi
σ	sigma
θ	theta

Acronym=>Meaning

Acronym	Meaning
CGI	Common Gateway Interface
DNET	DecisionNet
DSS	Decision Support System
GB	Gigabit
GUI	Graphical User Interface
HTML	Hyper Text Mark-Up Language
IDE	Integrated Development Environment
JIT	Just in Time
JVM	Java Virtual Machine
JDBC	Java DataBase Connectivity
JDK	Java Developer's Kit
KB	Kilobit
MB	Megabit
MHz	Megahertz
OS	Operating System
RAM	Random Access Memory
RMI	Remote Method Invocation
URL	Uniform Resource Locator
WWW	World Wide Web

APPENDIX B. TIPS FOR SYSTEM ADMINISTRATION

The system design is focused on adaptability beyond DecisionNet. It is able to display and manipulate any taxonomy that fits within the five variable, four dimension limit. This is achieved by passing parameters from an HTML page into each applet. A standard list of parameters is passed each time an applet is called. Variables that do not apply are passed as a blank space. An example of this generality is the URL parameter. The implementation is not tied to any URL or domain name. If the class files, data files, or database are transferred to another server it requires only a change to the HTML parameter.

The naming conventions and database design of DecisionNet were left untouched. The variable names internal to the Java code are in some cases specific to DecisionNet. This is transparent to the user because the actual values are fed from the HTML page. Each parameter is discussed in detail in Appendix C. Appropriate values are defined were applicable.

Each level of the taxonomy has its own data file. This was done to logically separate the taxonomy levels and to speed retrieval. The data files must conform to the standard format. The *parent* term starts at character position 1, followed by the *child* term at character position 39. The child term must be followed by a line break. This seemingly arbitrary number is two characters past the longest entry in the taxonomy. In the event that a taxonomy contains strings longer than 38 characters, the code of Grep.class would need to be altered and recompiled. The data files are stored as .txt files, and must be located in a WWW accessible directory along with the Java class files.

The Java code was developed in the Microsoft Visual J++ integrated development environment (IDE). If it becomes necessary to change any of the Java source code it should be done through the IDE. The Dnet project file (Dnet.dsw) contains

links to all of the Dnet source files. The files can be recompiled as a project by selecting the 'Rebuild All' function from the 'Build' pull down menu. The development platform was a Pentium (166 Mhz) with 64 MB of RAM. The operating system (OS) was Windows NT Workstation 4.0. The primary Web browser was Microsoft Internet Explorer 3.02. Netscape 3.0 and HotJava were used for test and evaluation only.

APPENDIX C. APPLET PARAMETERS

- PARAM NAME=title_input VALUE="DecisionNet Registration Applet" This is the screen title that appears in the top left corner of the taxonomy screen. This may be changed at will.

- PARAM NAME=LeftButtonLabel_input VALUE="Submit for Registration" There are only three acceptable values for this paramater.

  - Dnet_Reg.class requires the string "Submit for Registration".
  - Dnet_Search.class requires the string "Define Search Weights".
  - Dnet_Browse.class requires the string "Return to DecisionNet".

  These strings are used to tailor the taxonomy object based on the applet that called it. If anything other than these strings is used, the left button of the taxonomy screen will not function.

- PARAM NAME=ProviderID_Input VALUE="corgnati" This paramater is captured from a CGI script and inserted into HTML. Is a system defined user name. It is used by the registration applet.

- PARAM NAME=ConsumerID_Input VALUE=" " This paramater is captured from a CGI script and inserted into HTML. This is a system defined user name. It is used by the search applet.

- PARAM NAME=TechID_Input VALUE="14" This paramater is captured from a CGI script and inserted into HTML. It is an index to track producers with multiple technologies registered.

- PARAM NAME=TechName_Input VALUE="Test" This paramater is captured from a CGI script and inserted into HTML. This is whatever the producer wants it to be.

- PARAM NAME=tObjectType_Input VALUE="Solver" This parameter is captured from a CGI script and inserted into HTML. This is a relic from the original implimentation. It is included only to maintain integrity with the existing database and CGI scripts. The acceptable values for DecisionNet are:

  - ALL
  - Algorithm
  - Data Set
  - Decision Support System

- Example
- Model Schema
- Modeling Language
- Solver

- PARAM NAME=tURL_Input VALUE="http://131.120.39.60" This parameter is captured from a CGI script and inserted into HTML. It is the URL of the producer's technology.

- PARAM NAME=ExcInd_Input VALUE="Independent" This parameter is captured from a CGI script and inserted into HTML. This is a flag used by AMPL technologies. It is included only to maintain integrity with the existing database and CGI scripts.

- PARAM NAME=Purpose_Input VALUE="none" This parameter is captured from a CGI script and inserted into HTML. It is a narrative description of the technology.

- PARAM NAME=Comments_Input VALUE="none" This parameter is captured from a CGI script and inserted into HTML. It is a narrative that allows for any additional information.

- PARAM NAME=ProviderID_db_Input VALUE =" " This parameter represents the field name in the associated database. It is passed to the CGI scripts where required.

- PARAM NAME=ConsumerID_db_Input VALUE =" " This parameter represents the field name in the associated database. It is passed to the CGI scripts where required.

- PARAM NAME=TechID_db_Input VALUE =" " This parameter represents the field name in the associated database. It is passed to the CGI scripts where required.

- PARAM NAME=TechName_db_Input VALUE =" " This parameter represents the field name in the associated database. It is passed to the CGI scripts where required.

- PARAM NAME=tObjectType_db_Input VALUE =" " This parameter represents the field name in the associated database. It is passed to the CGI scripts where required.

- PARAM NAME=tURL_db_Input VALUE =" " This parameter represents the field name in the associated database. It is passed to the CGI scripts where required.

- PARAM NAME=ExcInd_db_Input VALUE =" " This parameter represents the field name in the associated database. It is passed to the CGI scripts where required.

- PARAM NAME=Purpose_db_Input VALUE =" " This parameter represents the field name in the associated database. It is passed to the CGI scripts where required.

- PARAM NAME=Comments_db_Input VALUE =" " This parameter represents the field name in the associated database. It is passed to the CGI scripts where required.

- PARAM NAME=V1_Label_Input VALUE="Problem Area" This is the label that will appear on the first button. It is also the string that will be searched for if the button is pressed. It is a parent term in the first level taxonomy data file. It should only be changed if using a taxonomy other than DecisionNet.

- PARAM NAME=V2_Label_Input VALUE="Functional Area" This is the label that will appear on the second button. It is also the string that will be searched for if the button is pressed. It is a parent term in the first level taxonomy data file. It should only be changed if using a taxonomy other than DecisionNet.

- PARAM NAME=V3_Label_Input VALUE="Industry Type" This is the label that will appear on the third button. It is also the string that will be searched for if the button is pressed. It is a parent term in the first level taxonomy data file. It should only be changed if using a taxonomy other than DecisionNet.

- PARAM NAME=V4_Label_Input VALUE="Organization Type" This is the label that will appear on the fourth button. It is also the string that will be searched for if the button is pressed. It is a parent term in the first level taxonomy data file. It should only be changed if using a taxonomy other than DecisionNet.

- PARAM NAME=V5_Label_Input VALUE="Solution Method" This is the label that will appear on the fifth button. It is also the string that will be searched for if the button is pressed. It is a parent term in the first level taxonomy data file. It should only be changed if using a taxonomy other than DecisionNet.

- PARAM NAME=Protocol_Input VALUE="http" This is the default protocol. This should not be changed.

- PARAM NAME=Data_URL_Input VALUE="131.120.39.60" This is the URL of the Java code as well as the taxonomy data files. It should be changed to reflect to actual location of those files.

- PARAM NAME=DB_URL_Input VALUE="131.120.39.63" This is the URL of the DecisionNet database and CGI scripts. It should be changed to reflect to actual location of those files.

- PARAM NAME=Tax_L1_FileLoc_Input VALUE="/Dnet/Tax_L1.txt" This is the directory location and file name of the taxonomy level 1 data file. It is relative to the WWW accessible root. These should be changed to reflect a new taxonomy or location.

- PARAM NAME=Tax_L2_FileLoc_Input VALUE="/Dnet/Tax_L2.txt" This is the directory location and file name of the taxonomy level 2 data file. It is relative to the WWW accessible root. These should be changed to reflect a new taxonomy or location.

- PARAM NAME=Tax_L3_FileLoc_Input VALUE="/Dnet/Tax_L3.txt" This is the directory location and file name of the taxonomy level 3 data file. It is relative to the WWW accessible root. These should be changed to reflect a new taxonomy or location.

- PARAM NAME=Tax_L4_FileLoc_Input VALUE="/Dnet/Tax_L4.txt" This is the directory location and file name of the taxonomy level 4 data file. It is relative to the WWW accessible root. These should be changed to reflect a new taxonomy or location.

- PARAM NAME=Reg_Script_Loc_Input VALUE= "/cgi-win/dnet/provider/regteca.exe" This is the directory location and file name of the registration script. It is relative to the WWW accessible root. It should be changed to reflect a new script or location.

- PARAM NAME=Search_Engine_Loc_Input VALUE=" " This is the directory location and file name of the search algorithm. It is relative to the WWW accessible root. It should be changed to reflect a new script or location.

- PARAM NAME=Default_Weight_Input VALUE=" " This paramater is used by the search algorith. It should be left at 0.2 to given even weight to all five variables. If fewer than 5 variables were utilized then it must be adjusted.

- PARAM NAME=Default_C_Deg_Input VALUE=" " This paramater is used by the search algorithm. It is set 0.8. It can be adjusted if further research indicates a better default.

- PARAM NAME=Default_P_Deg_Input VALUE=" " This paramater is used by the search algorithm. It is set 0.4. It can be adjusted if further research indicates a better default.

- PARAM NAME=Default_S_Deg_Input VALUE=" " This paramater is used by the search algorithm. It is set 0.6. It can be adjusted if further research indicates a better default.

- PARAM NAME=V1_cgi_label_Input VALUE="tProblemArea" This parameter is specific to the DecisionNet database. It should be changed to reflect the actual field name of the first variable of a new database.

- PARAM NAME=V1_value_default_Input VALUE="ALL" Default value for the initial display.

- PARAM NAME=V2_cgi_label_Input VALUE="tFunctionalArea" This parameter is specific to the DecisionNet database. It should be changed to reflect the actual field name of the second variable of a new database.

- PARAM NAME=V2_value_default_Input VALUE="ALL" Default value for the initial display.

- PARAM NAME=V3_cgi_label_Input VALUE="tIndType" This parameter is specific to the DecisionNet database. It should be changed to reflect the actual field name of the third variable of a new database.

- PARAM NAME=V3_value_default_Input VALUE="ALL" Default value for the initial display.

- PARAM NAME=V4_cgi_label_Input VALUE="tOrgType" This parameter is specific to the DecisionNet database. It should be changed to reflect the actual field name of the fourth variable of a new database.

- PARAM NAME=V4_value_default_Input VALUE="ALL" Default value for the initial display.

- PARAM NAME=V5_cgi_label_Input VALUE="tSolutionMethod" This parameter is specific to the DecisionNet database. It should be changed to reflect the actual field name of the fifth variable of a new database.

- PARAM NAME=V5_value_default_Input VALUE="ALL" Default value for the initial display.

APPENDIX D. BROWSER APPLET

```java
//**********************************************************************
// DNET_Browse.java:    Applet
// The DNET_Browse class is the applet that allows producers and
// consumers to view and explore the full multi-dimensional taxonomy
//
// Copyright by:  LT Chris Corgnati for Thesis work with Prof Bhargava
//
// Last modified:  1300 08 Aug 1997  version 1.1
//**********************************************************************
import java.applet.*;
import java.awt.*;


public class DNET_Browse extends Applet {

    //An HTML page must feed the following parameters to this applet
    String   title_Param;String message_Param;
    String   LeftButtonLabel_Param;

    String   ProviderID_Param;String ConsumerID_Param;
    String   TechID_Param;String TechName_Param;
    String   tObjectType_Param;
    String   tURL_Param;
    String   ExcInd_Param;
    String   Purpose_Param;String Comments_Param;

    String ProviderID_db_Param;String ConsumerID_db_Param;
    String   TechID_db_Param;String TechName_db_Param;
    String   tObjectType_db_Param;
    String   tURL_db_Param;
    String   ExcInd_db_Param;
    String   Purpose_db_Param;String Comments_db_Param;

    String   V1_Label_Param;String V2_Label_Param;
    String   V3_Label_Param;String V4_Label_Param;
    String   V5_Label_Param;

    String   Protocol_Param;String Data_URL_Param;
    String   DB_URL_Param;int Port_Param;
    String   Tax_L1_FileLoc_Param;String Tax_L2_FileLoc_Param;
```

51

```java
    String  Tax_L3_FileLoc_Param;String Tax_L4_FileLoc_Param;
    String  Reg_Script_Loc_Param;String Search_Engine_Loc_Param;

    String  Default_Weight_Param;String Default_C_Deg_Param;
    String  Default_P_Deg_Param;String Default_S_Deg_Param;

    String  V1_cgi_label_Param;String V1_value_default_Param;
    String  V2_cgi_label_Param;String V2_value_default_Param;
    String  V3_cgi_label_Param;String V3_value_default_Param;
    String  V4_cgi_label_Param;String V4_value_default_Param;
    String  V5_cgi_label_Param;String V5_value_default_Param;

// DNET Class Constructor
    public DNET_Browse() {
    }


// APPLET INFO SUPPORT:
    public String getAppletInfo()
    {
        return "Name: DNET\r\n" +
               "Author:  LT Christopher M. Corgnati\r\n" +
               "Created with Microsoft Visual J++ Version 1.1";
    }


//------------------------------------------------------------------
    public void init()  {

    this.title_Param = getParameter("title_input");
    this.LeftButtonLabel_Param =
        getParameter("LeftButtonLabel_input");
    this.ProviderID_Param = getParameter("ProviderID_Input");
    this.ConsumerID_Param = getParameter("ConsumerID_Input");
    this.TechID_Param = getParameter("TechID_Input");
    this.TechName_Param = getParameter("TechName_Input");
    this.tObjectType_Param = getParameter("tObjectType_Input");
    this.tURL_Param = getParameter("tURL_Input");
    this.ExcInd_Param = getParameter("ExcInd_Input");
    this.Purpose_Param = getParameter("Purpose_Input");
    this.Comments_Param = getParameter("Comments_Input");

    this.ProviderID_db_Param = getParameter("ProviderID_db_Input");
    this.ConsumerID_db_Param = getParameter("ConsumerID_db_Input");
```

52

```
this.TechID_db_Param = getParameter("TechID_db_Input");
this.TechName_db_Param = getParameter("TechName_db_Input");
this.tObjectType_db_Param = getParameter("tObjectType_db_Input");
this.tURL_db_Param = getParameter("tURL_db_Input");
this.ExcInd_db_Param = getParameter("ExcInd_db_Input");
this.Purpose_db_Param = getParameter("Purpose_db_Input");
this.Comments_db_Param = getParameter("Comments_db_Input");

this.V1_Label_Param = getParameter("V1_Label_Input");
this.V2_Label_Param = getParameter("V2_Label_Input");
this.V3_Label_Param = getParameter("V3_Label_Input");
this.V4_Label_Param = getParameter("V4_Label_Input");
this.V5_Label_Param = getParameter("V5_Label_Input");

this.Protocol_Param = getParameter("Protocol_Input");
this.Data_URL_Param = getParameter("Data_URL_Input");
this.DB_URL_Param = getParameter("DB_URL_Input");
Port_Param = 80;
this.Tax_L1_FileLoc_Param = getParameter("Tax_L1_FileLoc_Input");
this.Tax_L2_FileLoc_Param = getParameter("Tax_L2_FileLoc_Input");
this.Tax_L3_FileLoc_Param = getParameter("Tax_L3_FileLoc_Input");
this.Tax_L4_FileLoc_Param = getParameter("Tax_L4_FileLoc_Input");
this.Reg_Script_Loc_Param = getParameter("Reg_Script_Loc_Input");
this.Search_Engine_Loc_Param =
    getParameter("Search_Engine_Loc_Input");
this.Default_Weight_Param = getParameter("Default_Weight_Input");
this.Default_C_Deg_Param = getParameter("Default_C_Deg_Input");
this.Default_P_Deg_Param = getParameter("Default_P_Deg_Input");
this.Default_S_Deg_Param = getParameter("Default_S_Deg_Input");

this.V1_cgi_label_Param = getParameter("V1_cgi_label_Input");
this.V1_value_default_Param =
    getParameter("V1_value_default_Input");
this.V2_cgi_label_Param = getParameter("V2_cgi_label_Input");
this.V2_value_default_Param =
    getParameter("V2_value_default_Input");
this.V3_cgi_label_Param = getParameter("V3_cgi_label_Input");
this.V3_value_default_Param =
    getParameter("V3_value_default_Input");
this.V4_cgi_label_Param = getParameter("V4_cgi_label_Input");
this.V4_value_default_Param =
    getParameter("V4_value_default_Input");
```

```
        this.V5_cgi_label_Param = getParameter("V5_cgi_label_Input");
        this.V5_value_default_Param =
            getParameter("V5_value_default_Input");

//  This calls the Taxonomy screen.  Message can be changed here
        Frame f3 = new Frame();

        Taxonomy T = new Taxonomy(f3,title_Param,
            "Please use this browser to explore this multi-variable,
            multi-dimensional taxonomy. Use the \n"
            +"'Return' button to cancel the applet.  A double click on an
            item will reveal any items below \n"
            +"it in the Taxonomy.  It will also highlight your choice in
            the boxer above the lists. \n"
            +"This Applet is written in Java By LT Chris Corgnati.\n",
            LeftButtonLabel_Param,ProviderID_Param,TechID_Param,
            TechName_Param,tObjectType_Param,tURL_Param,ExcInd_Param,
            Purpose_Param,Comments_Param,ConsumerID_Param,V1_Label_Param,
            V2_Label_Param,V3_Label_Param,V4_Label_Param,V5_Label_Param,
            Protocol_Param,Data_URL_Param,DB_URL_Param,Port_Param,
            Tax_L1_FileLoc_Param,Tax_L2_FileLoc_Param,
            Tax_L3_FileLoc_Param,Tax_L4_FileLoc_Param,
            Reg_Script_Loc_Param,Search_Engine_Loc_Param,
            Default_Weight_Param,Default_C_Deg_Param,Default_P_Deg_Param,
            Default_S_Deg_Param,
            V1_cgi_label_Param,V1_value_default_Param,
            V2_cgi_label_Param,V2_value_default_Param,
            V3_cgi_label_Param,V3_value_default_Param,
            V4_cgi_label_Param,V4_value_default_Param,
            V5_cgi_label_Param,V5_value_default_Param,
            ConsumerID_db_Param,ProviderID_db_Param,
            TechID_db_Param,TechName_db_Param,tObjectType_db_Param,
            tURL_db_Param,ExcInd_db_Param,Purpose_db_Param,
            Comments_db_Param);

        T.resize(700,700);
        T.show();
    }
}
```

APPENDIX E. REGISTRATION APPLET

```
//*********************************************************************
// DNET_Reg.java:    Applet
// The DNET_Reg class is the applet that allows producers to register
// thier technologies with the full multi-dimensional taxonomy
//
// Copyright by:  LT Chris Corgnati for Thesis work with Prof Bhargava
//
// Last modified:  1300 08 Aug 1997   version 1.1
//*********************************************************************
import java.applet.*;
import java.awt.*;

public class DNET_Reg extends Applet {

    //An HTML page must feed the following parameters to this applet
    String   title_Param;String message_Param;
    String   LeftButtonLabel_Param;

    String   ProviderID_Param;String ConsumerID_Param;
    String   TechID_Param;String TechName_Param;
    String   tObjectType_Param;
    String   tURL_Param;
    String   ExcInd_Param;
    String   Purpose_Param;String Comments_Param;

    String ProviderID_db_Param;String ConsumerID_db_Param;
    String   TechID_db_Param;String TechName_db_Param;
    String   tObjectType_db_Param;
    String   tURL_db_Param;
    String   ExcInd_db_Param;
    String   Purpose_db_Param;String Comments_db_Param;

    String   V1_Label_Param;String V2_Label_Param;
    String   V3_Label_Param;String V4_Label_Param;
    String   V5_Label_Param;

    String   Protocol_Param;String Data_URL_Param;
    String   DB_URL_Param;int Port_Param;
    String   Tax_L1_FileLoc_Param;String Tax_L2_FileLoc_Param;
```

```java
    String   Tax_L3_FileLoc_Param;String Tax_L4_FileLoc_Param;
    String   Reg_Script_Loc_Param;String Search_Engine_Loc_Param;

    String   Default_Weight_Param;String Default_C_Deg_Param;
    String   Default_P_Deg_Param;String Default_S_Deg_Param;

    String   V1_cgi_label_Param;String V1_value_default_Param;
    String   V2_cgi_label_Param;String V2_value_default_Param;
    String   V3_cgi_label_Param;String V3_value_default_Param;
    String   V4_cgi_label_Param;String V4_value_default_Param;
    String   V5_cgi_label_Param;String V5_value_default_Param;


// Constructor Code
    public DNET_Reg() {
    }


// APPLET INFO SUPPORT:
    public String getAppletInfo()
    {
        return "Name: DNET\r\n" +
               "Author:  LT Christopher M. Corgnati\r\n" +
               "Created with Microsoft Visual J++ Version 1.1";
    }

//-----------------------------------------------------------------
    public void init()   {

    this.title_Param = getParameter("title_input");
    this.LeftButtonLabel_Param =
        getParameter("LeftButtonLabel_input");
    this.ProviderID_Param = getParameter("ProviderID_Input");
    this.ConsumerID_Param = getParameter("ConsumerID_Input");
    this.TechID_Param = getParameter("TechID_Input");
    this.TechName_Param = getParameter("TechName_Input");
    this.tObjectType_Param = getParameter("tObjectType_Input");
    this.tURL_Param = getParameter("tURL_Input");
    this.ExcInd_Param = getParameter("ExcInd_Input");
    this.Purpose_Param = getParameter("Purpose_Input");
    this.Comments_Param = getParameter("Comments_Input");

    this.ProviderID_db_Param = getParameter("ProviderID_db_Input");
```

56

```
this.ConsumerID_db_Param = getParameter("ConsumerID_db_Input");
this.TechID_db_Param = getParameter("TechID_db_Input");
this.TechName_db_Param = getParameter("TechName_db_Input");
this.tObjectType_db_Param = getParameter("tObjectType_db_Input");
this.tURL_db_Param = getParameter("tURL_db_Input");
this.ExcInd_db_Param = getParameter("ExcInd_db_Input");
this.Purpose_db_Param = getParameter("Purpose_db_Input");
this.Comments_db_Param = getParameter("Comments_db_Input");

this.V1_Label_Param = getParameter("V1_Label_Input");
this.V2_Label_Param = getParameter("V2_Label_Input");
this.V3_Label_Param = getParameter("V3_Label_Input");
this.V4_Label_Param = getParameter("V4_Label_Input");
this.V5_Label_Param = getParameter("V5_Label_Input");

this.Protocol_Param = getParameter("Protocol_Input");
this.Data_URL_Param = getParameter("Data_URL_Input");
this.DB_URL_Param = getParameter("DB_URL_Input");
Port_Param = 80;
this.Tax_L1_FileLoc_Param = getParameter("Tax_L1_FileLoc_Input");
this.Tax_L2_FileLoc_Param = getParameter("Tax_L2_FileLoc_Input");
this.Tax_L3_FileLoc_Param = getParameter("Tax_L3_FileLoc_Input");
this.Tax_L4_FileLoc_Param = getParameter("Tax_L4_FileLoc_Input");
this.Reg_Script_Loc_Param = getParameter("Reg_Script_Loc_Input");
this.Search_Engine_Loc_Param =
    getParameter("Search_Engine_Loc_Input");
this.Default_Weight_Param = getParameter("Default_Weight_Input");
this.Default_C_Deg_Param = getParameter("Default_C_Deg_Input");
this.Default_P_Deg_Param = getParameter("Default_P_Deg_Input");
this.Default_S_Deg_Param = getParameter("Default_S_Deg_Input");

this.V1_cgi_label_Param = getParameter("V1_cgi_label_Input");
this.V1_value_default_Param =
    getParameter("V1_value_default_Input");
this.V2_cgi_label_Param = getParameter("V2_cgi_label_Input");
this.V2_value_default_Param =
    getParameter("V2_value_default_Input");
this.V3_cgi_label_Param = getParameter("V3_cgi_label_Input");
this.V3_value_default_Param =
    getParameter("V3_value_default_Input");
this.V4_cgi_label_Param = getParameter("V4_cgi_label_Input");
this.V4_value_default_Param =
```

```
                  getParameter("V4_value_default_Input");
          this.V5_cgi_label_Param = getParameter("V5_cgi_label_Input");
          this.V5_value_default_Param =
                  getParameter("V5_value_default_Input");

//   This calls the Taxonomy screen.  Message can be changed here
      Frame f1 = new Frame();

      Taxonomy T = new Taxonomy(f1,title_Param,
              "Please use this browser to classify and register objects
              within the system. \n"
              +"Use the 'Submit' button to register your object.  Use the
              'Return' button to cancel \n"
              +"the applet.  A double click on an item will reveal any items
              below it in the Taxonomy. \n"
              +"It will also highlight your choice in the boxes above the
              lists. \n"
              +"This Applet is written in Java By LT Chris Corgnati.\n",
              LeftButtonLabel_Param,ProviderID_Param,TechID_Param,
              TechName_Param,tObjectType_Param,tURL_Param,ExcInd_Param,
              Purpose_Param,Comments_Param,ConsumerID_Param,V1_Label_Param,
              V2_Label_Param,V3_Label_Param,V4_Label_Param,V5_Label_Param,
              Protocol_Param,Data_URL_Param,DB_URL_Param,Port_Param,
              Tax_L1_FileLoc_Param,Tax_L2_FileLoc_Param,
              Tax_L3_FileLoc_Param,Tax_L4_FileLoc_Param,
              Reg_Script_Loc_Param,Search_Engine_Loc_Param,
              Default_Weight_Param,Default_C_Deg_Param,Default_P_Deg_Param,
              Default_S_Deg_Param,
              V1_cgi_label_Param,V1_value_default_Param,
              V2_cgi_label_Param,V2_value_default_Param,
              V3_cgi_label_Param,V3_value_default_Param,
              V4_cgi_label_Param,V4_value_default_Param,
              V5_cgi_label_Param,V5_value_default_Param,
              ConsumerID_db_Param,ProviderID_db_Param,
              TechID_db_Param,TechName_db_Param,tObjectType_db_Param,
              tURL_db_Param,ExcInd_db_Param,Purpose_db_Param,
              Comments_db_Param);

      T.resize(700,700);
      T.show();
      }
}
```

APPENDIX F. SEARCH APPLET

```
//***********************************************************************
// DNET_Search.java:    Applet
// The DNET_Search class is the applet that controls the inputs to the
// search algorithm and displays the results
//
// Copyright by:  LT Chris Corgnati for Thesis work with Prof Bhargava
//
// Last modified:  1350  08 Aug 1997 Version 1.1
//***********************************************************************
import java.applet.*;
import java.awt.*;


public class DNET_Search extends Applet {

    //An HTML page must feed the following parameters to this applet
    String  title_Param;String message_Param;
    String  LeftButtonLabel_Param;

    String  ProviderID_Param;String ConsumerID_Param;
    String  TechID_Param;String TechName_Param;
    String  tObjectType_Param;
    String  tURL_Param;
    String  ExcInd_Param;
    String  Purpose_Param;String Comments_Param;

    String ProviderID_db_Param;String ConsumerID_db_Param;
    String  TechID_db_Param;String TechName_db_Param;
    String  tObjectType_db_Param;
    String  tURL_db_Param;
    String  ExcInd_db_Param;
    String  Purpose_db_Param;String Comments_db_Param;

    String  V1_Label_Param;String V2_Label_Param;
    String  V3_Label_Param;String V4_Label_Param;
    String  V5_Label_Param;

    String  Protocol_Param;String Data_URL_Param;
    String  DB_URL_Param;int Port_Param;
    String  Tax_L1_FileLoc_Param;String Tax_L2_FileLoc_Param;
```

59

```java
    String   Tax_L3_FileLoc_Param;String Tax_L4_FileLoc_Param;
    String   Reg_Script_Loc_Param;String Search_Engine_Loc_Param;

    String   Default_Weight_Param;String Default_C_Deg_Param;
    String   Default_P_Deg_Param;String Default_S_Deg_Param;

    String   V1_cgi_label_Param;String V1_value_default_Param;
    String   V2_cgi_label_Param;String V2_value_default_Param;
    String   V3_cgi_label_Param;String V3_value_default_Param;
    String   V4_cgi_label_Param;String V4_value_default_Param;
    String   V5_cgi_label_Param;String V5_value_default_Param;


// Constructor Code
    public DNET_Search()     {
    }


// APPLET INFO SUPPORT:
    public String getAppletInfo()
    {
        return "Name: DNET\r\n" +
               "Author:  LT Christopher M. Corgnati\r\n" +
               "Created with Microsoft Visual J++ Version 1.1";
    }
//-----------------------------------------------------------------
public void init()  {

    this.title_Param = getParameter("title_input");
    this.LeftButtonLabel_Param =
        getParameter("LeftButtonLabel_input");
    this.ProviderID_Param = getParameter("ProviderID_Input");
    this.ConsumerID_Param = getParameter("ConsumerID_Input");
    this.TechID_Param = getParameter("TechID_Input");
    this.TechName_Param = getParameter("TechName_Input");
    this.tObjectType_Param = getParameter("tObjectType_Input");
    this.tURL_Param = getParameter("tURL_Input");
    this.ExcInd_Param = getParameter("ExcInd_Input");
    this.Purpose_Param = getParameter("Purpose_Input");
    this.Comments_Param = getParameter("Comments_Input");

    this.ProviderID_db_Param = getParameter("ProviderID_db_Input");
    this.ConsumerID_db_Param = getParameter("ConsumerID_db_Input");
```

60

```
this.TechID_db_Param = getParameter("TechID_db_Input");
this.TechName_db_Param = getParameter("TechName_db_Input");
this.tObjectType_db_Param = getParameter("tObjectType_db_Input");
this.tURL_db_Param = getParameter("tURL_db_Input");
this.ExcInd_db_Param = getParameter("ExcInd_db_Input");
this.Purpose_db_Param = getParameter("Purpose_db_Input");
this.Comments_db_Param = getParameter("Comments_db_Input");

this.V1_Label_Param = getParameter("V1_Label_Input");
this.V2_Label_Param = getParameter("V2_Label_Input");
this.V3_Label_Param = getParameter("V3_Label_Input");
this.V4_Label_Param = getParameter("V4_Label_Input");
this.V5_Label_Param = getParameter("V5_Label_Input");

this.Protocol_Param = getParameter("Protocol_Input");
this.Data_URL_Param = getParameter("Data_URL_Input");
this.DB_URL_Param = getParameter("DB_URL_Input");
Port_Param = 80;
this.Tax_L1_FileLoc_Param = getParameter("Tax_L1_FileLoc_Input");
this.Tax_L2_FileLoc_Param = getParameter("Tax_L2_FileLoc_Input");
this.Tax_L3_FileLoc_Param = getParameter("Tax_L3_FileLoc_Input");
this.Tax_L4_FileLoc_Param = getParameter("Tax_L4_FileLoc_Input");
this.Reg_Script_Loc_Param = getParameter("Reg_Script_Loc_Input");
this.Search_Engine_Loc_Param =
    getParameter("Search_Engine_Loc_Input");
this.Default_Weight_Param = getParameter("Default_Weight_Input");
this.Default_C_Deg_Param = getParameter("Default_C_Deg_Input");
this.Default_P_Deg_Param = getParameter("Default_P_Deg_Input");
this.Default_S_Deg_Param = getParameter("Default_S_Deg_Input");

this.V1_cgi_label_Param = getParameter("V1_cgi_label_Input");
this.V1_value_default_Param =
    getParameter("V1_value_default_Input");
this.V2_cgi_label_Param = getParameter("V2_cgi_label_Input");
this.V2_value_default_Param =
    getParameter("V2_value_default_Input");
this.V3_cgi_label_Param = getParameter("V3_cgi_label_Input");
this.V3_value_default_Param =
    getParameter("V3_value_default_Input");
this.V4_cgi_label_Param = getParameter("V4_cgi_label_Input");
this.V4_value_default_Param =
    getParameter("V4_value_default_Input");
```

```
            this.V5_cgi_label_Param = getParameter("V5_cgi_label_Input");
            this.V5_value_default_Param =
                getParameter("V5_value_default_Input");

    //  This calls the Taxonomy screen.  Message can be changed here
            Frame f2 = new Frame();

            Taxonomy T = new Taxonomy(f2,title_Param,
                "Welcome to the Search Engine.  Please use this applet to
                specify your preferences \n"
                +"and objectives for the search algorithm. Press 'Define
                Search Weights' to move to \n"
                +"the next screen.  Press 'Return' to cancel the applet.
                A double click on an item will \n"
                +"reveal any items below it in the Taxonomy.  It will also
                highlight your choice in the top boxes. \n"
                +"This Applet is written in Java By LT Chris Corgnati.\n",
                LeftButtonLabel_Param,ProviderID_Param,TechID_Param,
                TechName_Param,tObjectType_Param,tURL_Param,ExcInd_Param,
                Purpose_Param,Comments_Param,ConsumerID_Param,V1_Label_Param,
                V2_Label_Param,V3_Label_Param,V4_Label_Param,V5_Label_Param,
                Protocol_Param,Data_URL_Param,DB_URL_Param,Port_Param,
                Tax_L1_FileLoc_Param,Tax_L2_FileLoc_Param,
                Tax_L3_FileLoc_Param,Tax_L4_FileLoc_Param,
                Reg_Script_Loc_Param,Search_Engine_Loc_Param,
                Default_Weight_Param,Default_C_Deg_Param,Default_P_Deg_Param,
                Default_S_Deg_Param,
                V1_cgi_label_Param,V1_value_default_Param,
                V2_cgi_label_Param,V2_value_default_Param,
                V3_cgi_label_Param,V3_value_default_Param,
                V4_cgi_label_Param,V4_value_default_Param,
                V5_cgi_label_Param,V5_value_default_Param,
                ConsumerID_db_Param,ProviderID_db_Param,
                TechID_db_Param,TechName_db_Param,tObjectType_db_Param,
                tURL_db_Param,ExcInd_db_Param,Purpose_db_Param,
                Comments_db_Param);

            T.resize(700,700);
            T.show();
        }
    }
```

APPENDIX G. TAXONOMY

```java
// The Taxonomy class is the main interface for all of the Dnet
// applets
// Copyright by:  LT Chris Corgnati for Thesis work with
// Prof Bhargava
// Last modified:                           Version 1.2
//   1300  12 Aug 1997
//------------------------------------------------------------------
import java.awt.*;
import java.io.*;
import java.net.*;
import java.applet.*;

public class Taxonomy extends Frame {

    //Flags that pass control internal to this object
    private static final int cancel = 0;
    private static final int submit = 1;
    private static final int search = 27;

    private static final int V1_button_token = 2;
    private static final int V2_button_token = 3;
    private static final int V3_button_token = 4;
    private static final int V4_button_token = 5;
    private static final int V5_button_token = 6;

    private static final int V1_token = 7;
//  private static final int V1_2D_token = 8;   //future use
//  private static final int V1_3D_token = 9;   //future use
//  private static final int V1_4D_token = 10;  //future use

    private static final int V2_token = 11;
//  private static final int V2_2D_token = 12;  //future use
//  private static final int V2_3D_token = 13;  //future use
//  private static final int V2_4D_token = 14;  //future use

    private static final int V3_token = 15;
    private static final int V3_2D_token = 16;  //future use
//  private static final int V3_3D_token = 17;  //future use
//  private static final int V3_4D_token = 18;  //future use
```

```java
    private static final int V4_token = 19;
//  private static final int V4_2D_token = 20;   //future use
//  private static final int V4_3D_token = 21;   //future use
//  private static final int V4_4D_token = 22;   //future use

    private static final int V5_token = 23;
    private static final int V5_2D_token = 24;
    private static final int V5_3D_token = 25;
    private static final int V5_4D_token = 26;

    // Value strings are set to the parameter strings passed from HTML
    String ProviaerID_Value = new String(ProviderID_Param);
    String ConsumerID_Value = new String(ConsumerID_Param);
    String TechID_Value = new String(TechID_Param);
    String TechName_Value = new String(TechName_Param);
    String tObjectType_Value = new String(tObjectType_Param);
    String tURL_Value = new String(tURL_Param);
    String ExcInd_Value = new String(ExcInd_Param);
    String Purpose_Value = new String(Purpose_Param);
    String Comments_Value = new String(Comments_Param);

    //these represent the database column names-passed from HTML
    String ConsumerID_db_value = new String(ConsumerID_db);
    String ProviderID_db_value = new String(ProviderID_db);
    String TechID_db_value = new String(TechID_db);
    String TechName_db_value = new String(TechName_db);
    String tObjectType_db_value = new String(tObjectType_db);
    String tURL_db_value = new String(tURL_db);
    String ExcInd_db_value = new String(ExcInd_db);
    String Purpose_db_value = new String(Purpose_db);
    String Comments_db_value = new String(Comments_db);

    //button labels
    String V1_Label = new String(V1_Label_Param);
    String V2_Label = new String(V2_Label_Param);
    String V3_Label = new String(V3_Label_Param);
    String V4_Label = new String(V4_Label_Param);
    String V5_Label = new String(V5_Label_Param);

    //Networking variables and file locations
    String Protocol = new String(Protocol);
```

```java
String Data_File_Home = new String(Data_URL);
String Database_Home = new String(DB_URL);
int port = Port;
String Tax_L1 = new String(Tax_L1_FileLoc);
String Tax_L2 = new String(Tax_L2_FileLoc);
String Tax_L3 = new String(Tax_L3_FileLoc);
String Tax_L4 = new String(Tax_L4_FileLoc);
String cgi_regteca = new String(Reg_Script_Loc);
String Search_Engine_Loc = new String(Search_Engine_Loc);

//default search values
String  Default_Weight = new String(Default_Weight_Param);
String  Default_C_Deg = new String(Default_C_Deg_Param);
String  Default_P_Deg = new String(Default_P_Deg_Param);
String  Default_S_Deg = new String(Default_S_Deg_Param);

//cgi and database names, and default values for initial display
String V1_cgi_label = new String(V1_cgi_label);
String V1_Value = new String(V1_value_default);
String V2_cgi_label = new String(V2_cgi_label);
String V2_Value = new String(V2_value_default);
String V3_cgi_label = new String(V3_cgi_label);
String V3_Value = new String(V3_value_default);
String V4_cgi_label = new String(V4_cgi_label);
String V4_Value = new String(V4_value_default);
String V5_cgi_label = new String(V5_cgi_label);
String V5_Value = new String(V5_value_default);

private int List_Token;     //shows the active list

Button   Submit,Cancel,V1_Button,V2_Button,
         V3_Button,V4_Button,V5_Button;

MultiLineLabel Narrative;

List            Level_1,Level_2,Level_3,Level_4;

TextField       V1_Field,V1_2D_Field,
                 V1_3D_Field,V1_4D_Field;
TextField       V2_Field,V2_2D_Field,
                 V2_3D_Field,V2_4D_Field;
TextField       V3_Field,V3_2D_Field,
```

```
                    V3_3D_Field,V3_4D_Field;
        TextField       V4_Field,V4_2D_Field,
                    V4_3D_Field,V4_4D_Field;
        TextField       V5_Field,V5_2D_Field,
                    V5_3D_Field,V5_4D_Field;


        TextField       Current_Field,Message_Field;


        GridBagLayout gridbag = new GridBagLayout();


        InputStream input;


        String querystring;
        // value is used to differentiate which applet is calling
        String button_label = new String(Submit_Button_Label);
//-----------------------------------------------------------------
// Constuctor Code-instances of taxonomy must call it with this info

        public Taxonomy(Frame f,String title,String message,
            String Submit_Button_Label,
            String ProviderID_Param,String TechID_Param,
            String TechName_Param,String tObjectType_Param,
            String tURL_Param,String ExcInd_Param,String Purpose_Param,
            String Comments_Param,String ConsumerID_Param,
            String V1_Label_Param,String V2_Label_Param,
            String V3_Label_Param,String V4_Label_Param,
            String V5_Label_Param,String Protocol,
            String Data_URL,String DB_URL,int Port,String Tax_L1_FileLoc,
            String Tax_L2_FileLoc,String Tax_L3_FileLoc,
            String Tax_L4_FileLoc,
            String Reg_Script_Loc,String Search_Engine_Loc,
            String Default_Weight_Param,String  Default_C_Deg_Param,
            String Default_P_Deg_Param,String Default_S_Deg_Param,
            String V1_cgi_label,String V1_value_default,
            String V2_cgi_label,String V2_value_default,
            String V3_cgi_label,String V3_value_default,
            String V4_cgi_label,String V4_value_default,
            String V5_cgi_label,String V5_value_default,
            String ConsumerID_db,String ProviderID_db,
            String TechID_db,String TechName_db,String tObjectType_db,
            String tURL_db,String ExcInd_db,String Purpose_db,
            String Comments_db) {
```

```
        super(title);

//------------------------------------------------------------------
// Panel 1-Buttons
        Cancel = new Button ("Return to DecisionNet");
        Submit = new Button (Submit_Button_Label);

        Message_Field = new TextField(30);
        Message_Field.setEditable(false);

        Panel p1 = new Panel();
        p1.setLayout(gridbag);

        constrain(p1,Submit,0,0,1,1,GridBagConstraints.BOTH,
            GridBagConstraints.CENTER,0.0,0.0,10,10,10,10);
        constrain(p1,Message_Field,1,0,1,1,GridBagConstraints.BOTH,
            GridBagConstraints.CENTER,0.0,0.0,10,10,10,10);
        constrain(p1,Cancel,2,0,1,1,GridBagConstraints.BOTH,
            GridBagConstraints.CENTER,0.0,0.0,10,10,10,10);
//------------------------------------------------------------------
// Panel 2-Instructions
        Narrative = new MultiLineLabel(message,1);
        Panel p2 = new Panel();
        p2.setLayout(gridbag);
        constrain(p2,Narrative,0,0,1,1,GridBagConstraints.HORIZONTAL,
            GridBagConstraints.CENTER,0.0,0.0,0,0,0,0);
//------------------------------------------------------------------
// Panel 3-Selection Display
        V1_Button = new Button    (V1_Label);
        V2_Button = new Button    (V2_Label);
        V3_Button = new Button    (V3_Label);
        V4_Button = new Button    (V4_Label);
        V5_Button = new Button    (V5_Label);

        V1_Field = new TextField(18);
            V1_Field.setText(V1_Value);
            V1_Field.setEditable(false);
        V1_2D_Field = new TextField(18);
            V1_2D_Field.setEditable(false);
        V1_3D_Field = new TextField(18);
            V1_3D_Field.setEditable(false);
        V1_4D_Field = new TextField(18);
```

67

```
                V1_4D_Field.setEditable(false);

    V2_Field = new TextField(18);
        V2_Field.setText(V2_Value);
        V2_Field.setEditable(false);
    V2_2D_Field = new TextField(18);
        V2_2D_Field.setEditable(false);
    V2_3D_Field = new TextField(18);
        V2_3D_Field.setEditable(false);
    V2_4D_Field = new TextField(18);
        V2_4D_Field.setEditable(false);

    V3_Field = new TextField(18);
        V3_Field.setText(V3_Value);
        V3_Field.setEditable(false);
    V3_2D_Field = new TextField(18);
        V3_2D_Field.setEditable(false);
    V3_3D_Field = new TextField(18);
        V3_3D_Field.setEditable(false);
    V3_4D_Field = new TextField(18);
        V3_4D_Field.setEditable(false);

    V4_Field = new TextField(18);
        V4_Field.setText(V4_Value);
        V4_Field.setEditable(false);
    V4_2D_Field = new TextField(18);
        V4_2D_Field.setEditable(false);
    V4_3D_Field = new TextField(18);
        V4_3D_Field.setEditable(false);
    V4_4D_Field = new TextField(18);
        V4_4D_Field.setEditable(false);

    V5_Field = new TextField(18);
        V5_Field.setText(V5_Value);
        V5_Field.setEditable(false);
    V5_2D_Field = new TextField(18);
        V5_2D_Field.setEditable(false);
    V5_3D_Field = new TextField(18);
        V5_3D_Field.setEditable(false);
    V5_4D_Field = new TextField(18);
        V5_4D_Field.setEditable(false);
```

```
Panel p3 = new Panel();
p3.setLayout(gridbag);

constrain(p3,V1_Button,1,0,1,1,GridBagConstraints.HORIZONTAL,
    GridBagConstraints.CENTER,0.0,0.0,1,1,1,1);
constrain(p3,V2_Button,1,1,1,1,GridBagConstraints.HORIZONTAL,
    GridBagConstraints.CENTER,0.0,0.0,1,1,1,1);
constrain(p3,V3_Button,1,2,1,1,GridBagConstraints.HORIZONTAL,
    GridBagConstraints.CENTER,0.0,0.0,1,1,1,1);
constrain(p3,V4_Button,1,3,1,1,GridBagConstraints.HORIZONTAL,
    GridBagConstraints.CENTER,0.0,0.0,1,1,1,1);
constrain(p3,V5_Button,1,4,1,1,GridBagConstraints.HORIZONTAL,
    GridBagConstraints.CENTER,0.0,0.0,1,1,1,1);

constrain(p3,V1_Field,2,0,1,1,GridBagConstraints.HORIZONTAL,
    GridBagConstraints.CENTER,0.0,0.0,1,1,1,1);
constrain(p3,V1_2D_Field,3,0,1,1,GridBagConstraints.HORIZONTAL,
    GridBagConstraints.CENTER,0.0,0.0,1,1,1,1);
constrain(p3,V1_3D_Field,4,0,1,1,GridBagConstraints.HORIZONTAL,
    GridBagConstraints.CENTER,0.0,0.0,1,1,1,1);
constrain(p3,V1_4D_Field,5,0,1,1,GridBagConstraints.HORIZONTAL,
    GridBagConstraints.CENTER,0.0,0.0,1,1,1,1);

constrain(p3,V2_Field,2,1,1,1,GridBagConstraints.HORIZONTAL,
    GridBagConstraints.CENTER,0.0,0.0,1,1,1,1);
constrain(p3,V2_2D_Field,3,1,1,1,GridBagConstraints.HORIZONTAL,
    GridBagConstraints.CENTER,0.0,0.0,1,1,1,1);
constrain(p3,V2_3D_Field,4,1,1,1,GridBagConstraints.HORIZONTAL,
    GridBagConstraints.CENTER,0.0,0.0,1,1,1,1);
constrain(p3,V2_4D_Field,5,1,1,1,GridBagConstraints.HORIZONTAL,
    GridBagConstraints.CENTER,0.0,0.0,1,1,1,1);

constrain(p3,V3_Field,2,2,1,1,GridBagConstraints.HORIZONTAL,
    GridBagConstraints.CENTER,0.0,0.0,1,1,1,1);
constrain(p3,V3_2D_Field,3,2,1,1,GridBagConstraints.HORIZONTAL,
    GridBagConstraints.CENTER,0.0,0.0,1,1,1,1);
constrain(p3,V3_3D_Field,4,2,1,1,GridBagConstraints.HORIZONTAL,
    GridBagConstraints.CENTER,0.0,0.0,1,1,1,1);
constrain(p3,V3_4D_Field,5,2,1,1,GridBagConstraints.HORIZONTAL,
    GridBagConstraints.CENTER,0.0,0.0,1,1,1,1);

constrain(p3,V4_Field,2,3,1,1,GridBagConstraints.HORIZONTAL,
```

```
                    GridBagConstraints.CENTER,0.0,0.0,1,1,1,1);
         constrain(p3,V4_2D_Field,3,3,1,1,GridBagConstraints.HORIZONTAL,
                    GridBagConstraints.CENTER,0.0,0.0,1,1,1,1);
         constrain(p3,V4_3D_Field,4,3,1,1,GridBagConstraints.HORIZONTAL,
                    GridBagConstraints.CENTER,0.0,0.0,1,1,1,1);
         constrain(p3,V4_4D_Field,5,3,1,1,GridBagConstraints.HORIZONTAL,
                    GridBagConstraints.CENTER,0.0,0.0,1,1,1,1);

         constrain(p3,V5_Field,2,4,1,1,GridBagConstraints.HORIZONTAL,
                    GridBagConstraints.CENTER,0.0,0.0,1,1,1,1);
         constrain(p3,V5_2D_Field,3,4,1,1,GridBagConstraints.HORIZONTAL,
                    GridBagConstraints.CENTER,0.0,0.0,1,1,1,1);
         constrain(p3,V5_3D_Field,4,4,1,1,GridBagConstraints.HORIZONTAL,
                    GridBagConstraints.CENTER,0.0,0.0,1,1,1,1);
         constrain(p3,V5_4D_Field,5,4,1,1,GridBagConstraints.HORIZONTAL,
                    GridBagConstraints.CENTER,0.0,0.0,1,1,1,1);
//-----------------------------------------------------------------
// Panel 4-Display boxes for hierarchical levels
         Current_Field = new TextField(30);
         Current_Field.setEditable(false);

         Level_1 = new List (12, false);
         Level_2 = new List (12, false);
         Level_3 = new List (12, false);
         Level_4 = new List (12, false);

      Panel p4 = new Panel();
      p4.setLayout(gridbag);

      constrain(p4,Current_Field,2,0,2,1,GridBagConstraints.HORIZONTAL,
          GridBagConstraints.CENTER,0.0,0.0,1,1,1,1);

      constrain(p4,Level_1,1,1,1,1,GridBagConstraints.HORIZONTAL,
          GridBagConstraints.CENTER,0.0,0.0,1,1,1,1);
      constrain(p4,Level_2,2,1,1,1,GridBagConstraints.HORIZONTAL,
          GridBagConstraints.CENTER,0.0,0.0,1,1,1,1);
      constrain(p4,Level_3,3,1,1,1,GridBagConstraints.HORIZONTAL,
          GridBagConstraints.CENTER,0.0,0.0,1,1,1,1);
      constrain(p4,Level_4,4,1,1,1,GridBagConstraints.HORIZONTAL,
          GridBagConstraints.CENTER,0.0,0.0,1,1,1,1);
//-----------------------------------------------------------------
// Add panels to the top level window and arrange the panels
```

70

```java
        this.setLayout(gridbag);

        constrain (this,p1,1,0,1,1,GridBagConstraints.NONE,
                GridBagConstraints.CENTER,0.0,0.0,1,1,1,1);
        constrain (this,p2,1,1,1,1,GridBagConstraints.NONE,
                GridBagConstraints.CENTER,0.0,0.0,1,1,1,1);
        constrain (this,p3,1,2,1,1,GridBagConstraints.NONE,
                GridBagConstraints.CENTER,0.0,0.0,4,4,4,4);
        constrain (this,p4,1,3,1,1,GridBagConstraints.NONE,
                GridBagConstraints.CENTER,0.0,0.0,4,4,4,4);
    }
//--------------------------------------------------------------------
// Event Handler
    public boolean action(Event event,Object arg) {
        if (event.target instanceof Button) {
          if (event.target == Submit) {
            if (((String)event.arg).equals("Submit for Registration"))
                answer(submit);
              else if (((String)event.arg).equals
                  ("Define Search Weights")) answer(search);
                else if (((String)event.arg).equals
                    ("Return to DecisionNet")) answer(cancel);
            }
          else if (event.target == V1_Button) answer(V1_button_token);
          else if (event.target == V2_Button) answer(V2_button_token);
          else if (event.target == V3_Button) answer(V3_button_token);
          else if (event.target == V4_Button) answer(V4_button_token);
          else if (event.target == V5_Button) answer(V5_button_token);
          else answer(cancel);
          return true;
        }
        if (event.target instanceof List) {
            if ((event.target == Level_1) & (List_Token == 111))
                answer(V1_token);
//          else if ((event.target == Level_2) & (List_Token == 111))
//              answer(V1_2D_token);
//          else if ((event.target == Level_3) & (List_Token == 111))
//              answer(V1_3D_token);
//          else if ((event.target == Level_4) & (List_Token == 111))
//              answer(V1_4D_token);

            else if ((event.target == Level_1) & (List_Token == 222))
```

```
                                answer(V2_token);
//              else if ((event.target == Level_2) & (List_Token == 222))
//                    answer(V2_2D_token);
//              else if ((event.target == Level_3) & (List_Token == 222))
//                    answer(V2_3D_token);
//              else if ((event.target == Level_4) & (List_Token == 222))
//                    answer(V2_4D_token);

                else if ((event.target == Level_1) & (List_Token == 333))
                      answer(V3_token);
                else if ((event.target == Level_2) & (List_Token == 333))
                      answer(V3_2D_token);
//              else if ((event.target == Level_3) & (List_Token == 333))
//                    answer(V3_3D_token);
//              else if ((event.target == Level_4) & (List_Token == 333))
//                    answer(V3_4D_token);

                else if ((event.target == Level_1) & (List_Token == 444))
                      answer(V4_token);
//              else if ((event.target == Level_2) & (List_Token == 444))
//                    answer(V4_2D_token);
//              else if ((event.target == Level_3) & (List_Token == 444))
//                    answer(V4_3D_token);
//              else if ((event.target == Level_4) & (List_Token == 444))
//                    answer(V4_4D_token);

                else if ((event.target == Level_1) & (List_Token == 555))
                      answer(V5_token);
                else if ((event.target == Level_2) & (List_Token == 555))
                      answer(V5_2D_token);
                else if ((event.target == Level_3) & (List_Token == 555))
                      answer(V5_3D_token);
                else if ((event.target == Level_4) & (List_Token == 555))
                      answer(V5_4D_token);

                else answer(99);
                return true;
        }
            else return false;
    }
//----------------------------------------------------------------
    protected void answer(int answer) {
```

```
        switch(arswer) {
        case cancel:cancel_handler();break;
        case submit:submit_handler();break;
        case search:search_handler();break;

        case V1_button_token:V1_button_handler();break;
        case V2_button_token:V2_button_handler();break;
        case V3_button_token:V3_button_handler();break;
        case V4_button_token:V4_button_handler();break;
        case V5_button_token:V5_button_handler();break;

        case V1_token:V1_handler();break;
//      case V1_2D_token:V1_2D_handler();break;
//      case V1_3D_token:V1_3D_handler();break;
//      case V1_4D_token:V1_4D_handler();break;

        case V2_token:V2_handler();break;
//      case V2_2D_token:V2_2D_handler();break;
//      case V2_3D_token:V2_3D_handler();break;
//      case V2_4D_token:V2_4D_handler();break;

        case V3_token:V3_handler();break;
        case V3_2D_token:V3_2D_handler();break;
//      case V3_3D_token:V3_3D_handler();break;
//      case V3_4D_token:V3_4D_handler();break;

        case V4_token:V4_handler();break;
//      case V4_2D_token:V4_2D_handler();break;
//      case V4_3D_token:V4_3D_handler();break;
//      case V4_4D_token:V4_4D_handler();break;

        case V5_token:V5_handler();break;
        case V5_2D_token:V5_2D_handler();break;
        case V5_3D_token:V5_3D_handler();break;
        case V5_4D_token:V5_4D_handler();break;
        }
    }
//-----------------------------------------------------------------
    protected void cancel_handler() {
        this.hide();
        this.dispose();
    }
```

```
//-----------------------------------------------------------------
//used to call the search engine
//The cgi script dictates that the order be 1,2,5,3,4

    protected void search_handler() {
        Weights W = new Weights("DecisionNet Search Engine",
            "Please use this applet to specify your \n"
            +"preferences and objectives for the search algorithm.\n"
            +"Press 'Submit to Search Engine' to launch your search.\n"
            +"This Applet is written in Java By LT Chris Corgnati.\n"
            +"                        \n",
            ConsumerID_Value,tObjectType_Value,V1_Value,V2_Value,
            V5_Value,V3_Value,V4_Value,Protocol,Database_Home,
            port,Search_Engine_Loc,Default_Weight,Default_C_Deg,
            Default_P_Deg,Default_S_Deg,ConsumerID_db_value,
            tObjectType_db_value,V1_cgi_label,V2_cgi_label,
            V3_cgi_label,V4_cgi_label,V5_cgi_label);

        W.resize(700,700);
        W.show();

        this.hide();
        this.dispose();
    }
//-----------------------------------------------------------------
//used to call the registration cgi script
    public void submit_handler() {
//resets the querystring
        String querystring =  " ";

//These varibles need to be passed in from the HTML page to the applet
        QueryString Q = new QueryString(ProviderID_db_value,
                                        ProviderID_Value);
        Q.add(TechID_db_value,TechID_Value);
        Q.add(TechName_db_value,TechName_Value);
        Q.add(tObjectType_db_value,tObjectType_Value);

//These variables are user selections
//The cgi script dictates that the order be 1,2,5,3,4
        Q.add(V1_cgi_label,V1_Value);
        Q.add(V2_cgi_label,V2_Value);
        Q.add(V5_cgi_label,V5_Value);
```

74

```
        Q.add(V3 cgi_label,V3_Value);
        Q.add(V4_cgi_label,V4_Value);


//These varibles need to be passed in from the HTML page to the applet
        Q.add(tURL_db_value,tURL_Value);
        Q.add(ExcInd_db_value,ExcInd_Value);
        Q.add(Purpose_db_value,Purpose_Value);
        Q.add(Comments_db_value,Comments_Value);


//sets local string to return of QueryString method
        querystring = Q.query;


//attempt to establish a connection
        try {
            Message_Field.setText("Establishing a Network Connection");
            URL u = new URL(Protocol,Database_Home,port,cgi_regteca);
            URLConnection connection = u.openConnection();
            connection.setDoOutput(true);
            connection.setDoInput(true);
            connection.setAllowUserInteraction(false);

            //send data to cgi program
            DataOutputStream dos = new DataOutputStream
                (connection.getOutputStream());
            dos.writeBytes(querystring);
            dos.close();

             try {
                String thisLine;
                String results = (" ");

                DataInputStream DIS = new DataInputStream
                    (connection.getInputStream());
                    while ((thisLine = DIS.readLine()) != null) {
                        results = results + thisLine;
                    }
                DIS.close();

                //Open Dialog to display registration results
                Frame f4 = new Frame();
                Reg_Complete D = new Reg_Complete
                  (f4,"DecisionNet Registration Results",
```

```
                    "Registration is Complete. \n"
                    +"Thank you for using DecisionNet. \n"
                    +"Press 'Return' to go back to DecisionNet. \n"
                    +"                                          \n"
                    +"                                          \n",
                    "Return to DecisionNet");
                D.show();
                D.resize(300,250);
                this.hide();
                ┼his.dispose();

                Message_Field.setText("Search Complete");
            }

                //catch any exceptions
                catch (MalformedURLException e) {
                    Message_Field.setText(e.toString());
                }

                catch (IOException e) {
                    Message_Field.setText(e.toString());
                }

                catch (NullPointerException e) {
                    Message_Field.setText(e.toString());
                }
            }

//catch any exceptions
        catch (MalformedURLException e) {
          Message_Field.setText(e.toString());
          Frame f4 = new Frame();
          Reg_Complete D = new Reg_Complete
            (f4,"DecisionNet Registration Results",
            "Registration Failed(MalformedURLException). \n"
            +"If this continues, please contact DecisionNet. \n"
            +"Press the 'Return' button to go back to DecisionNet. \n"
            +"                                                    \n"
            +"                                                    \n",
            "Return to DecisionNet");
        D.show();
        D.resize(300,250);
```

```java
              this.hide();
              this.dispose();
          }

          catch (IOException e) {
            Message_Field.setText(e.toString());
            Frame f4 = new Frame();
            Reg_Complete D = new Reg_Complete
              (f4,"DecisionNet Registration Results",
              "Registration Failed(IOException). \n"
              +"If this continues, please contact DecisionNet. \n"
              +"Press the 'Return' button to go back to DecisionNet. \n"
              +"                                                  \n"
              +"                                                  \n",
              "Return to DecisionNet");
            D.show();
            D.resize(300,250);
            this.hide();
            this.dispose();
          }

          catch (NullPointerException e) {
            Message_Field.setText(e.toString());
            Frame f4 = new Frame();
            Reg_Complete D = new Reg_Complete
              (f4,"DecisionNet Registration Results",
              "Registration Failed(NullPointerException). \n"
              +"If this continues, please contact DecisionNet. \n"
              +"Press the 'Return' button to go back to DecisionNet. \n"
              +"                                                  \n"
              +"                                                  \n",
              "Return to DecisionNet");
            D.show();
            D.resize(300,250);
            this.hide();
            this.dispose();
          }
      }
//----------------------------------------------------------------
//----------------------------------------------------------------
    protected void V1_button_handler () {
        Current_Field.setText(V1_Label);
```

```
            List_Token = 111;
            button_handler(V1_Label);
        }
    //------------------------------------------------------------------
        protected void V1_handler () {
            Level_2.clear();Level_3.clear();Level_4.clear();
            V1_Field.setText(Level_1.getSelectedItem());
            V1_2D_Field.setText("");V1_3D_Field.setText("");
            V1_4D_Field.setText("");V1_Value=Level_1.getSelectedItem();
    //        level_handler(Protocol,Data_File_Home,Tax_L2,Level_1,Level_2);
        }
    //------------------------------------------------------------------
    // for future expansion to three dimensions of V1 type variable
    //  protected void V1_2D_handler () {
    //        Level_3.clear();Level_4.clear();
    //        V1_2D_Field.setText(Level_2.getSelectedItem());
    //        V1_3D_Field.setText("");V1_4D_Field.setText("");
    //        V1_Value=Level_2.getSelectedItem();
    //        level_handler(Protocol,Data_File_Home,Tax_L3,Level_1,Level_2);
    //  }
    //------------------------------------------------------------------
    // for future expansion to four dimensions of V1 type variable
    //  protected void V1_3D_handler () {
    //        Level_4.clear();
    //        V1_3D_Field.setText(Level_3.getSelectedItem());
    //        V1_4D_Field.setText("");
    //        V1_Value=Level_3.getSelectedItem();
    //        level_handler(Protocol,Data_File_Home,Tax_L4,Level_1,Level_2);
    //  }
    //------------------------------------------------------------------
    // for future expansion to four dimensions of V1 type variable
    //  protected void V1_4D_handler () {
    //        V1_4D_Field.setText(Level_4.getSelectedItem());
    //        V1_Value=Level_4.getSelectedItem();
    //  }
    //------------------------------------------------------------------
    //------------------------------------------------------------------
        protected void V2_button_handler () {
            Current_Field.setText(V2_Label);
            List_Token = 222;
            button_handler(V2_Label);
        }
```

```
//-------------- ----------------------------------------------------------
    protected void V2_handler () {
        Level_2.clear();Level_3.clear();Level_4.clear();
        V2_Field.setText(Level_1.getSelectedItem());
        V2_2D_Field.setText("");V2_3D_Field.setText("");
        V2_4D_Field.setText("");V2_Value=Level_1.getSelectedItem();
//      level_handler(Protocol,Data_File_Home,Tax_L2,Level_1,Level_2);
    }
//--------------------------------------------------------------------------
// for future expansion to three dimensions of V2 area variable
//  protected void V2_2D_handler () {
//      Level_3.clear();Level_4.clear();
//      V2_2D_Field.setText(Level_2.getSelectedItem());
//      V2_3D_Field.setText("");V2_4D_Field.setText("");
//      tV2Area_Value=Level_2.getSelectedItem();
//      level_handler(Protocol,Data_File_Home,Tax_L3,Level_1,Level_2);
//  }
//--------------------------------------------------------------------------
// for future expansion to four dimensions of V2 area variable
//  protected void V2_3D_handler () {
//      Level_4.clear();
//      V2_3D_Field.setText(Level_3.getSelectedItem());
//      V2_4D_Field.setText("");
//      tV2Area_Value=Level_3.getSelectedItem();
//      level_handler(Protocol,Data_File_Home,Tax_L4,Level_1,Level_2);
//  }
//--------------------------------------------------------------------------
// for future expansion to four dimensions of V2 area variable
//  protected void V2_4D_handler () {
//      V2_4D_Field.setText(Level_4.getSelectedItem());
//      tV2Area_Value=Level_4.getSelectedItem();
//  }
//--------------------------------------------------------------------------
//--------------------------------------------------------------------------
    protected void V3_button_handler () {
        Current_Field.setText(V3_Label);
        List_Token = 333;
        button_handler(V3_Label);
    }
//--------------------------------------------------------------------------
//--------------------------------------------------------------------------
    protected void V3_handler () {
```

```
            (Level_2.clear());Level_3.clear();Level_4.clear();
            V3_Field.setText(Level_1.getSelectedItem());
            V3_2D_Field.setText("");V3_3D_Field.setText("");
            V3_4D_Field.setText("");V3_Value=Level_1.getSelectedItem();
            level_handler(Protocol,Data_File_Home,Tax_L2,Level_1,Level_2);
        }
//--------------------------------------------------------------------
// for future expansion to three dimensions of V3 type variable
    protected void V3_2D_handler () {
            Level_3.clear();Level_4.clear();
            V3_2D_Field.setText(Level_2.getSelectedItem());
            V3_3D_Field.setText("");V3_4D_Field.setText("");
            V3_Value=Level_2.getSelectedItem();
            level_handler(Protocol,Data_File_Home,Tax_L3,Level_2,Level_3);
        }
//--------------------------------------------------------------------
// for future expansion to four dimensions of V3 type variable
//   protected void V3_3D_handler () {
//         Level_4.clear();
//         V3_3D_Field.setText(Level_3.getSelectedItem());
//         V3_4D_Field.setText("");V3_Value=Level_3.getSelectedItem();
//         level_handler(Protocol,Data_File_Home,Tax_L4,Level_3,Level_4);
//   }
//--------------------------------------------------------------------
// for future expansion to four dimensions of V3 type variable
//   protected void V3_4D_handler () {
//         V3_4D_Field.setText(Level_4.getSelectedItem());
//         V3_Value=Level_4.getSelectedItem();
//   }
//--------------------------------------------------------------------
//--------------------------------------------------------------------
    protected void V4_button_handler () {
            Current_Field.setText(V4_Label);
            List_Token = 444;
            button_handler(V4_Label);
        }
//--------------------------------------------------------------------
    protected void V4_handler () {
            (Level_2.clear());Level_3.clear();Level_4.clear();
            V4_Field.setText(Level_1.getSelectedItem());
            V4_2D_Field.setText("");V4_3D_Field.setText("");
            V4_4D_Field.setText("");V4_Value=Level_1.getSelectedItem();
```

```
            level_handler(Protocol,Data_File_Home,Tax_L2,Level_1,Level_2);
        }
//----------------------------------------------------------------
//  protected void V4_2D_handler () {
//        Level_3.clear();Level_4.clear();
//        V4_2D_Field.setText(Level_2.getSelectedItem());
//        V4_3D_Field.setText("");V4_4D_Field.setText("");
//        V4_Value=Level_2.getSelectedItem();
//        level_handler(Protocol,Data_File_Home,Tax_L3,Level_2,Level_3);
//  }
//----------------------------------------------------------------
// for future expansion to four dimensions of V4 type variable
//  protected void V4_3D_handler () {
//        Level_4.clear();
//        V4_3D_Field.setText(Level_3.getSelectedItem());
//        V4_4D_Field.setText("");V4_Value=Level_3.getSelectedItem();
//        level_handler(Protocol,Data_File_Home,Tax_L4,Level_3,Level_4);
//  }
//----------------------------------------------------------------
// for future expansion to four dimensions of V4 type variable
//  protected void V4_4D_handler () {
//        V4_4D_Field.setText(Level_4.getSelectedItem());
//        V4_Value=Level_4.getSelectedItem();
//  }
//----------------------------------------------------------------
//----------------------------------------------------------------
    protected void V5_button_handler () {
        Current_Field.setText(V5_Label);
        List_Token = 555;
        button_handler(V5_Label);
    }
//----------------------------------------------------------------
    protected void V5_handler () {
        (Level_2.clear());(Level_3.clear());(Level_4.clear());
        V5_Field.setText(Level_1.getSelectedItem());
        V5_2D_Field.setText("");V5_3D_Field.setText("");
        V5_4D_Field.setText("");
        V5_Value=Level_1.getSelectedItem();
        level_handler(Protocol,Data_File_Home,Tax_L2,Level_1,Level_2);
    }
//----------------------------------------------------------------
    protected void V5_2D_handler () {
```

```
            (Level_3.clear());(Level_4.clear());
            V5_2D_Field.setText(Level_2.getSelectedItem());
            V5_3D_Field.setText("");V5_4D_Field.setText("");
            V5_Value=Level_2.getSelectedItem();
            level_handler(Protocol,Data_File_Home,Tax_L3,Level_2,Level_3);
        }
//----------------------------------------------------------------
    protected void V5_3D_handler () {
            (Level_4.clear());
            V5_3D_Field.setText(Level_3.getSelectedItem());
            V5_4D_Field.setText("");V5_Value=Level_3.getSelectedItem();
            level_handler(Protocol,Data_File_Home,Tax_L4,Level_3,Level_4);
        }
//----------------------------------------------------------------
    protected void V5_4D_handler () {
            V5_4D_Field.setText(Level_4.getSelectedItem());
            V5_Value=Level_4.getSelectedItem();
        }
//----------------------------------------------------------------
//----------------------------------------------------------------
    protected void button_handler (String var_name) {
            Level_1.clear();Level_2.clear();
            Level_3.clear();Level_4.clear();
            try {
                URL url = new URL(Protocol,Data_File_Home,Tax_L1);
                URLConnection connection = url.openConnection();
                input = connection.getInputStream();
                Grep g = new Grep(var_name,Level_1,input);
            }
            catch (MalformedURLException e) {
                Current_Field.setText("URL Exception");
            }
            catch (IOException i) {
                Current_Field.setText("I/O Exception");
            }
        }
//----------------------------------------------------------------
    protected void level_handler(String Protocol,
        String Data_File_Home,String FileLocation,
        List Current,List Target) {
            try {
                URL url = new URL(Protocol,Data_File_Home,FileLocation);
```

```java
            URLConnection connection = url.openConnection();
            input = connection.getInputStream();
            Grep g = new Grep(Current.getSelectedItem(),Target,input);
        }
        catch (MalformedURLException e) {
            Current_Field.setText("URL Exception");
        }
        catch (IOException i) {
            Current_Field.setText("I/O Exception");
        }
    }
//----------------------------------------------------------------
//----------------------------------------------------------------
//These constrain functions are used to layout the components both
//within thier panels and the panels within the frame.
//(from Java in Nutshell)
    public void constrain (Container container,Component component,
        int grid_x,int grid_y,int grid_width,int grid_height,
        int fill,int anchor,double weight_x,double weight_y,
        int top,int left,int bottom,int right) {

        GridBagConstraints c = new GridBagConstraints();
        c.gridx = grid_x;c.gridy = grid_y;
        c.gridwidth = grid_width;c.gridheight = grid_height;
        c.fill = fill;c.anchor = anchor;
        c.weightx = weight_x;c.weighty = weight_y;
        if(top + bottom + left + right > 0)
            c.insets = new Insets(top,left,bottom,right);
        ((GridBagLayout)container.getLayout()).setConstraints
          (component,c);
        container.add(component);
    }
//----------------------------------------------------------------
    public void constrain (Container container,Component component,
        int grid_x,int grid_y,int grid_width,int grid_height) {
        constrain (container,component,grid_x,grid_y,grid_width,
        grid_height,GridBagConstraints.NONE,
        GridBagConstraints.NORTHWEST,0.0,0.0,0,0,0,0);
    }
//----------------------------------------------------------------
    public void constrain (Container container,Component component,
        int grid_x,int grid_y,int grid_width,int grid_height,
```

```
            int top,int left,int bottom,int right) {
        constrain (container,component,grid_x,grid_y,grid_width,
        grid_height,GridBagConstraints.NONE,
        GridBagConstraints.NORTHWEST,0.0,0.0,top,left,bottom,right);
    }
}
```

APPENDIX H. WEIGHTS

```
// The Weights class is the interface for both consumer search and
// producer registration
//
// Copyright by:  LT Chris Corgnati for Thesis work with Prof Bhargava
//
// Last modified:                                Version 1.1
//   1300 29 July 1997
//------------------------------------------------------------------
import java.awt.*;
import java.io.*;
import java.net.*;

public class Weights extends Frame {

    //Flags that pass control internal to this object
    Button   Search,Cancel;

    Label    Problem_Label,Functional_Label,Organization_Label,
             Industry_Label,Solution_Label,Weight_Col,Child_Col,
             Parent_Col,Sibling_Col;

    MultiLineLabel Narrative,Legend;

    TextField        Problem_Weight,Problem_C_Deg,
                      Problem_P_Deg,Problem_S_Deg;
    TextField        Functional_Weight,Functional_C_Deg,
                      Functional_P_Deg,Functional_S_Deg;
    TextField        Organization_Weight,Organization_C_Deg,
                      Organization_P_Deg,Organization_S_Deg;
    TextField        Industry_Weight,Industry_C_Deg,
                      Industry_P_Deg,Industry_S_Deg;
    TextField        Solution_Weight,Solution_C_Deg,
                      Solution_P_Deg,Solution_S_Deg;

    TextField        Message_Field;

    String           Default_Weight = new String(Default_Weight_Param);
    String           Default_C_Deg = new String(Default_C_Deg_Param);
    String           Default_P_Deg = new String(Default_P_Deg_Param);
```

```java
        String           Default_S_Deg = new String(Default_S_Deg_Param);

        GridBagLayout gridbag = new GridBagLayout();

//Variables used to create the querystring
    String ConsumerID = new String(ConsumerID_db);
    String tObjectType = new String(tObjectType_db);
    String tProblemArea = new String(tProblemArea_db);
    String tFunctionalArea = new String(tFunctionalArea_db);
    String tSolutionMethod = new String(tSolutionMethod_db);
    String tIndType = new String(tIndType_db);
    String tOrgType = new String(tOrgType_db);

//Variables passed from taxonomy object-will be submitted to search
    String ConsumerID_Value = new String(ConsumerID_Param);
    String tObjectType_Value = new String(tObjectType_Param);
    String tProblemArea_Value = new String(tProblemArea_Param);
    String tFunctionalArea_Value = new String(tFunctionalArea_Param);
    String tSolutionMethod_Value = new String(tSolutionMethod_Param);
    String tIndType_Value = new String(tIndType_Param);
    String tOrgType_Value = new String(tOrgType_Param);

//Network variables to connect to CGI programs
    String Protocol = new String(Protocol);
    String dnet1 = new String(DB_URL);
    int port = Port;
    //  cgi_indexed is the home of the delphi search engine script
    String cgi_indexed = new String(Search_Engine_Loc);
//------------------------------------------------------------------
// Constuctor Code-instances of taxonomy must call it with this info
   public Weights(String title,String message,String ConsumerID_Param,
       String tObjectType_Param,String tProblemArea_Param,
       String tFunctionalArea_Param,String tSolutionMethod_Param,
       String tIndType_Param,String tOrgType_Param,String Protocol,
       String DB_URL,int Port,String Search_Engine_Loc,
       String Default_Weight_Param,String Default_C_Deg_Param,
       String  Default_P_Deg_Param,String Default_S_Deg_Param,
       String ConsumerID_db,String tObjectType_db,
       String tProblemArea_db,String tFunctionalArea_db,
       String tSolutionMethod_db,String tIndType_db,
       String tOrgType_db) {
       super(title);
```

```
//------------------------------------------------------------------
// Panel 1-Instructions
        Search = new Button ("Submit to Search Engine");
        Cancel = new Button ("   Return to DecisionNet   ");

        Message_Field = new TextField(30);
        Message_Field.setEditable(false);

        Panel p1 = new Panel();
        p1.setLayout(gridbag);

        constrain(p1,Search,0,0,1,1,GridBagConstraints.BOTH,
            GridBagConstraints.CENTER,0.0,0.0,1,1,1,1);
        constrain(p1,Message_Field,1,0,1,1,GridBagConstraints.BOTH,
            GridBagConstraints.CENTER,0.0,0.0,1,1,1,1);
        constrain(p1,Cancel,2,0,1,1,GridBagConstraints.BOTH,
            GridBagConstraints.CENTER,0.0,0.0,1,1,1,1);
//------------------------------------------------------------------
// Panel 2-Instructions
        Narrative = new MultiLineLabel(message,1);

        Panel p2 = new Panel();
        p2.setLayout(gridbag);

        constrain(p2,Narrative,0,0,1,1,GridBagConstraints.HORIZONTAL,
            GridBagConstraints.CENTER,0.0,0.0,0,0,0,0);
//------------------------------------------------------------------
// Panel 3-Instructions
        Legend = new MultiLineLabel
        ("VW = Variable Weight (must sum to 1.0) \n"
        +" \n"
        +"CDM = Child Degradation Multiplier \n"
        +"PDM = Parent Degradation Multiplier \n"
        +"SDM = Sibling Degradation Multiplier \n"
        +"   \n"
        +"Default Values are shown. Advanced users may define \n"
        +"custom values here.  Degradation Multipliers must be in \n"
        +"the interval [0,1]. Higher values indicate a stronger
        preference \n"
        +"for the selected item only.  Lower values allow the search
        to \n"
        +"return more possible answers which may have a lower
```

```
        relevancy.",1);

        Panel p3 = new Panel();
        p3.setLayout(gridbag);

        constrain(p3,Legend,0,0,1,1,GridBagConstraints.HORIZONTAL,
            GridBagConstraints.CENTER,0.0,0.0,0,0,0,0);
//------------------------------------------------------------------
// Panel 3-Selection Display

    Weight_Col = new Label("VW");
    Child_Col = new Label("CDM");
    Parent_Col = new Label("PDM");
    Sibling_Col = new Label("SDM");


    Problem_Label = new Label("Problem Area");
    Functional_Label = new Label("Functional Area");
    Industry_Label = new Label("Industry Type");
    Organization_Label = new Label("Organization Type");
    Solution_Label = new Label("Solution Area");

    Problem_Weight = new TextField(Default_Weight,5);
      Problem_Weight.setEditable(true);
    Problem_C_Deg = new TextField(Default_C_Deg,5);
      Problem_C_Deg.setEditable(true);
    Problem_P_Deg = new TextField(Default_P_Deg,5);
      Problem_P_Deg.setEditable(true);
    Problem_S_Deg = new TextField(Default_S_Deg,5);
      Problem_S_Deg.setEditable(true);

    Functional_Weight = new TextField(Default_Weight,5);
      Functional_Weight.setEditable(true);
    Functional_C_Deg = new TextField(Default_C_Deg,5);
      Functional_C_Deg.setEditable(true);
    Functional_P_Deg = new TextField(Default_P_Deg,5);
      Functional_P_Deg.setEditable(true);
    Functional_S_Deg = new TextField(Default_S_Deg,5);
      Functional_S_Deg.setEditable(true);

    Industry_Weight = new TextField(Default_Weight,5);
      Industry_Weight.setEditable(true);
    Industry_C_Deg = new TextField(Default_C_Deg,5);
```

88

```
   Industry_C_Deg.setEditable(true);
 Industry_P_Deg = new TextField(Default_P_Deg,5);
   Industry_P_Deg.setEditable(true);
 Industry_S_Deg = new TextField(Default_S_Deg,5);
   Industry_S_Deg.setEditable(true);

 Organization_Weight = new TextField(Default_Weight,5);
   Organization_Weight.setEditable(true);
 Organization_C_Deg = new TextField(Default_C_Deg,5);
   Organization_C_Deg.setEditable(true);
 Organization_P_Deg = new TextField(Default_P_Deg,5);
   Organization_P_Deg.setEditable(true);
 Organization_S_Deg = new TextField(Default_S_Deg,5);
   Organization_S_Deg.setEditable(true);

 Solution_Weight = new TextField(Default_Weight,5);
   Solution_Weight.setEditable(true);
 Solution_C_Deg = new TextField(Default_C_Deg,5);
   Solution_C_Deg.setEditable(true);
 Solution_P_Deg = new TextField(Default_P_Deg,5);
   Solution_P_Deg.setEditable(true);
 Solution_S_Deg = new TextField(Default_S_Deg,5);
   Solution_S_Deg.setEditable(true);

Panel p5 = new Panel();
  p5.setLayout(gridbag);

constrain(p5,Weight_Col,1,0,1,1,GridBagConstraints.HORIZONTAL,
  GridBagConstraints.CENTER,0.0,0.0,1,1,1,1);
constrain(p5,Child_Col,2,0,1,1,GridBagConstraints.HORIZONTAL,
  GridBagConstraints.CENTER,0.0,0.0,1,1,1,1);
constrain(p5,Parent_Col,3,0,1,1,GridBagConstraints.HORIZONTAL,
  GridBagConstraints.CENTER,0.0,0.0,1,1,1,1);
constrain(p5,Sibling_Col,4,0,1,1,GridBagConstraints.HORIZONTAL,
  GridBagConstraints.CENTER,0.0,0.0,1,1,1,1);

constrain(p5,Problem_Label,0,1,1,1,GridBagConstraints.HORIZONTAL,
  GridBagConstraints.CENTER,0.0,0.0,1,1,1,1);
constrain(p5,Functional_Label,0,2,1,1,GridBagConstraints.HORIZONTAL,
  GridBagConstraints.CENTER,0.0,0.0,1,1,1,1);
constrain(p5,Industry_Label,0,3,1,1,GridBagConstraints.HORIZONTAL,
  GridBagConstraints.CENTER,0.0,0.0,1,1,1,1);
```

```
constrain(p5,Organization_Label,0,4,1,1,
  GridBagConstraints.HORIZONTAL,GridBagConstraints.CENTER,
  0.0,0.0,1,1,1,1);
constrain(p5,Solution_Label,0,5,1,1,GridBagConstraints.HORIZONTAL,
  GridBagConstraints.CENTER,0.0,0.0,1,1,1,1);
constrain(p5,Problem_Weight,1,1,1,1,GridBagConstraints.HORIZONTAL,
  GridBagConstraints.CENTER,0.0,0.0,1,1,1,1);
constrain(p5,Problem_C_Deg,2,1,1,1,GridBagConstraints.HORIZONTAL,
  GridBagConstraints.CENTER,0.0,0.0,1,1,1,1);
constrain(p5,Problem_P_Deg,3,1,1,1,GridBagConstraints.HORIZONTAL,
  GridBagConstraints.CENTER,0.0,0.0,1,1,1,1);
constrain(p5,Problem_S_Deg,4,1,1,1,GridBagConstraints.HORIZONTAL,
  GridBagConstraints.CENTER,0.0,0.0,1,1,1,1);

constrain(p5,Functional_Weight,1,2,1,1,
  GridBagConstraints.HORIZONTAL,GridBagConstraints.CENTER,
  0.0,0.0,1,1,1,1);
constrain(p5,Functional_C_Deg,2,2,1,1,GridBagConstraints.HORIZONTAL,
  GridBagConstraints.CENTER,0.0,0.0,1,1,1,1);
constrain(p5,Functional_P_Deg,3,2,1,1,GridBagConstraints.HORIZONTAL,
  GridBagConstraints.CENTER,0.0,0.0,1,1,1,1);
constrain(p5,Functional_S_Deg,4,2,1,1,GridBagConstraints.HORIZONTAL,
  GridBagConstraints.CENTER,0.0,0.0,1,1,1,1);

constrain(p5,Industry_Weight,1,3,1,1,GridBagConstraints.HORIZONTAL,
  GridBagConstraints.CENTER,0.0,0.0,1,1,1,1);
constrain(p5,Industry_C_Deg,2,3,1,1,GridBagConstraints.HORIZONTAL,
  GridBagConstraints.CENTER,0.0,0.0,1,1,1,1);
constrain(p5,Industry_P_Deg,3,3,1,1,GridBagConstraints.HORIZONTAL,
  GridBagConstraints.CENTER,0.0,0.0,1,1,1,1);
constrain(p5,Industry_S_Deg,4,3,1,1,GridBagConstraints.HORIZONTAL,
  GridBagConstraints.CENTER,0.0,0.0,1,1,1,1);

constrain(p5,Organization_Weight,1,4,1,1,
  GridBagConstraints.HORIZONTAL,GridBagConstraints.CENTER,
  0.0,0.0,1,1,1,1);
constrain(p5,Organization_C_Deg,2,4,1,1,
  GridBagConstraints.HORIZONTAL,GridBagConstraints.CENTER,
  0.0,0.0,1,1,1,1);
constrain(p5,Organization_P_Deg,3,4,1,1,
  GridBagConstraints.HORIZONTAL,GridBagConstraints.CENTER,
  0.0,0.0,1,1,1,1);
```

```
    constrain(p5,Organization_S_Deg,4,4,1,1,
      GridBagConstraints.HORIZONTAL,GridBagConstraints.CENTER,
      0.0,0.0,1,1,1,1);

    constrain(p5,Solution_Weight,1,5,1,1,GridBagConstraints.HORIZONTAL,
      GridBagConstraints.CENTER,0.0,0.0,1,1,1,1);
    constrain(p5,Solution_C_Deg,2,5,1,1,GridBagConstraints.HORIZONTAL,
      GridBagConstraints.CENTER,0.0,0.0,1,1,1,1);
    constrain(p5,Solution_P_Deg,3,5,1,1,GridBagConstraints.HORIZONTAL,
      GridBagConstraints.CENTER,0.0,0.0,1,1,1,1);
    constrain(p5,Solution_S_Deg,4,5,1,1,GridBagConstraints.HORIZONTAL,
      GridBagConstraints.CENTER,0.0,0.0,1,1,1,1);
//-------------------------------------------------------------------
// Add panels to the top level window and arrange the panels
        this.setLayout(gridbag);

        constrain (this,p1,0,0,1,1,GridBagConstraints.NONE,
                GridBagConstraints.CENTER,0.0,0.0,1,1,1,1);
        constrain (this,p2,0,1,1,1,GridBagConstraints.NONE,
                GridBagConstraints.CENTER,0.0,0.0,1,1,1,1);
        constrain (this,p3,0,2,1,1,GridBagConstraints.NONE,
                GridBagConstraints.CENTER,0.0,0.0,4,4,4,4);
        constrain (this,p5,0,4,1,1,GridBagConstraints.NONE,
                GridBagConstraints.CENTER,0.0,0.0,4,4,4,4);
    }
//-------------------------------------------------------------------
    public boolean action(Event event,Object arg) {
            if (event.target instanceof Button) {
                if (event.target == Cancel) {
                    this.hide();
                    this.dispose();
                }

                if (event.target == Search) {
                //resets the querystring
                String querystring =  " ";

                //These varibles need to be passed in from the HTML
                //page to the applet
                QueryString Q = new QueryString(ConsumerID,
                ConsumerID_Value);
                Q.add(tObjectType,tObjectType_Value);
```

91

```java
//These variables are user selections
Q.add(tProblemArea,tProblemArea_Value);
Q.add(tFunctionalArea,tFunctionalArea_Value);
Q.add(tSolutionMethod,tSolutionMethod_Value);
Q.add(tIndType,tIndType_Value);
Q.add(tOrgType,tOrgType_Value);

String PW = new String(Problem_Weight.getText());
Q.add("Problem_Weight",PW);
String PCD = new String(Problem_C_Deg.getText());
Q.add("Problem_C_Deg",PCD);
String PPD = new String(Problem_P_Deg.getText());
Q.add("Problem_P_Deg",PPD);
String PSD = new String(Problem_S_Deg.getText());
Q.add("Problem_S_Deg",PSD);

String FW = new String(Functional_Weight.getText());
Q.add("Functional_Weight",FW);
String FCD = new String(Functional_C_Deg.getText());
Q.add("Functional_C_Deg",FCD);
String FPD = new String(Functional_P_Deg.getText());
Q.add("Functional_P_Deg",FPD);
String FSD = new String(Functional_S_Deg.getText());
Q.add("Functional_S_Deg",FSD);

String IW = new String(Industry_Weight.getText());
Q.add("Industry_Weight",IW);
String ICD = new String(Industry_C_Deg.getText());
Q.add("Industry_C_Deg",ICD);
String IPD = new String(Industry_P_Deg.getText());
Q.add("Industry_P_Deg",IPD);
String ISD = new String(Industry_S_Deg.getText());
Q.add("Industry_S_Deg",ISD);

String OW = new String(Organization_Weight.getText());
Q.add("Organization_Weight",OW);
String OCD = new String(Organization_C_Deg.getText());
Q.add("Organization_C_Deg",OCD);
String OPD = new String(Organization_P_Deg.getText());
Q.add("Organization_P_Deg",OPD);
String OSD = new String(Organization_S_Deg.getText());
```

```
Q.add("Organization_S_Deg",OSD);

String SW = new String(Solution_Weight.getText());
Q.add("Solution_Weight",SW);
String SCD = new String(Solution_C_Deg.getText());
Q.add("Solution_C_Deg",SCD);
String SPD = new String(Solution_P_Deg.getText());
Q.add("Solution_P_Deg",SPD);
String SSD = new String(Solution_S_Deg.getText());
Q.add("Solution_S_Deg",SSD);

//sets local string to return of QueryString method
querystring = Q.query;

//attempt to establish a connection
try {
Message_Field.setText("Establishing Network
Connection....");
URL u = new URL(Protocol,dnet1,port,cgi_indexed);
URLConnection connection = u.openConnection();
Message_Field.setText("Searching DecisionNet...");
connection.setDoOutput(true);
connection.setDoInput(true);
connection.setAllowUserInteraction(false);

//send data to cgi program
DataOutputStream dos = new DataOutputStream
(connection.getOutputStream());
dos.writeBytes(querystring);
dos.close();

    //read in CGI response (string of text)
    try {
    String thisLine;
    String results = (" ");

    DataInputStream DIS = new DataInputStream
    (connection.getInputStream());
        while ((thisLine = DIS.readLine()) != null) {
            results = results + thisLine;
        }
```

```
        DIS.close();
        Message_Field.setText("Search Complete");

        //call results display screen which will parse the
        //input stream and show it
        Frame f4 = new Frame();
        Search_Results S = new Search_Results
        (f4,"DecisionNet Search Results",
        "Results of your search. \n"
        +"Thank you for using DecisionNet. \n"
        +"Press the 'Return' button to go back to
        DecisionNet. \n"
        +"                                          \n"
        +"                                          \n",
        "Return to DecisionNet",results);
        S.show();
        S.resize(700,700);
        this.hide();
        this.dispose();
        }

        //catch any exceptions
        catch (MalformedURLException e) {
            Message_Field.setText(e.toString());
        }

        catch (IOException e) {
            Message_Field.setText(e.toString());
        }

        catch (NullPointerException e) {
            Message_Field.setText(e.toString());
        }
}

//catch any exceptions
catch (MalformedURLException e) {
Message_Field.setText(e.toString());
Frame f4 = new Frame();
Reg_Complete D = new Reg_Complete
  (f4,"DecisionNet Search Results",
  "Search Engine Failed(MalformedURLException). \n"
```

94

```java
        +"If this continues, please contact DecisionNet. \n"
        +"Press the 'Return' button to go back to
        DecisionNet. \n"
        +"                                              \n"
        +"                                              \n",
        "Return to DecisionNet");
    D.show();
    D.resize(300,250);
    this.hide();
    this.dispose();
    }

    catch (IOException e) {
    Message_Field.setText("Connection Failed");
    Frame f4 = new Frame();
    Reg_Complete D = new Reg_Complete
    (f4,"DecisionNet Search Results",
        "Search Engine Failed(IOException). \n"
        +"If this continues, please contact DecisionNet. \n"
        +"Press the 'Return' button to go back to
        DecisionNet. \n"
        +"                                              \n"
        +"                                              \n",
        "Return to DecisionNet");
    D.show();
    D.resize(300,250);
    this.hide();
    this.dispose();
    }

    catch (NullPointerException e) {
    Message_Field.setText(e.toString());
    Frame f4 = new Frame();
    Reg_Complete D = new Reg_Complete
        (f4,"DecisionNet Search Results",
        "Search Engine Failed(NullPointerException). \n"
        +"If this continues, please contact DecisionNet. \n"
        +"Press the 'Return' button to go back to
        DecisionNet. \n"
        +"                                              \n"
        +"                                              \n",
        "Return to DecisionNet");
```

95

```
                D.show();
                D.resize(300,250);
                this.hide();
                this.dispose();

                }
            }
            return true;
        }
        else return false;
    }
//-------------------------------------------------------------------
//These constrain functions are used to layout the components both
//within thier panels and the panels within the frame.
//(from Java in Nutshell)
    public void constrain (Container container,Component component,
        int grid_x,int grid_y,int grid_width,int grid_height,
        int fill,int anchor,double weight_x,double weight_y,
        int top,int left,int bottom,int right) {

        GridBagConstraints c = new GridBagConstraints();
        c.gridx = grid_x;c.gridy = grid_y;
        c.gridwidth = grid_width;c.gridheight = grid_height;
        c.fill = fill;c.anchor = anchor;
        c.weightx = weight_x;c.weighty = weight_y;
        if(top + bottom + left + right > 0)
            c.insets = new Insets(top,left,bottom,right);
        ((GridBagLayout)container.getLayout()).setConstraints
          (component,c);
        container.add(component);
    }
//-------------------------------------------------------------------
    public void constrain (Container container,Component component,
        int grid_x,int grid_y,int grid_width,int grid_height) {
        constrain (container,component,grid_x,grid_y,grid_width,
        grid_height,GridBagConstraints.NONE,
        GridBagConstraints.NORTHWEST,0.0,0.0,0,0,0,0);
    }
//-------------------------------------------------------------------
    public void constrain (Container container,Component component,
        int grid_x,int grid_y,int grid_width,int grid_height,
        int top,int left,int bottom,int right) {
```

```
        constrain (container,component,grid_x,grid_y,grid_width,
        grid_height,GridBagConstraints.NONE,
        GridBagConstraints.NORTHWEST,0.0,0.0,top,left,bottom,right);
    }
}
```

APPENDIX I. SEARCH RESULTS

```java
// The Search Results class is used to display the data returned from
// the cgi script search engine
//
// Copyright by:  LT Chris Corgnati for Thesis work with Prof Bhargava
//
// Last modified:  1400  29 July 1997 Version 1.0
//------------------------------------------------------------------
//------------------------------------------------------------------
import java.awt.*;
import java.io.*;
import java.util.*;

public class Search_Results extends Dialog {

    // Control variables are internal to this object
    private static final int Cancel = 1;

    protected Button cancel;

    protected MultiLineLabel label;

    protected TextArea display;

    protected String line;
    protected String[] lines;

    protected int num_lines;

    GridBagLayout gridbag = new GridBagLayout();

    // Constructor code
    public Search_Results(Frame f,String title,String message,
        String Cancel_label,String results)
    {
        super(f,"DecisionNet",false);
        this.setLayout(new BorderLayout(40,60));

//------------------------------------------------------------------
        // Creation and layout of GUI components
```

99

```
        Panel p1 = new Panel();
        p1.setLayout(gridbag);
        label = new MultiLineLabel(message,1);

        constrain(p1,label,0,0,1,1,GridBagConstraints.HORIZONTAL,
            GridBagConstraints.CENTER,0.0,0.0,4,4,4,4);

        Panel p2 = new Panel();
        p2.setLayout(gridbag);
        display = new TextArea(23,70);

        //this code parses the input stream by the character &
        //into an array of strings to be displayed
        StringTokenizer t = new StringTokenizer(results,"&");
        num_lines = t.countTokens();
        lines = new String[num_lines];
        for(int i = 0;i < num_lines;i++)
            lines[i] = (t.nextToken()+" \n");
        for(int i = 0;i < num_lines;i++)display.appendText(lines[i]);

        constrain(p2,display,0,0,1,1,GridBagConstraints.HORIZONTAL,
            GridBagConstraints.CENTER,0.0,0.0,4,4,4,4);

        Panel p3 = new Panel();
        p3.setLayout(gridbag);
        cancel=new Button (Cancel_label);

        constrain(p3,cancel,0,0,1,1,GridBagConstraints.HORIZONTAL,
            GridBagConstraints.CENTER,0.0,0.0,4,4,4,4);

        this.show();
//-----------------------------------------------------------------
        // Arrange the panels
        this.setLayout(gridbag);
        // Add panels to the top level window
        constrain (this,p1,1,0,1,1,GridBagConstraints.BOTH,
                GridBagConstraints.CENTER,0.0,0.0,3,3,3,3);
        constrain (this,p2,1,1,1,1,GridBagConstraints.HORIZONTAL,
                GridBagConstraints.CENTER,0.0,0.0,3,3,3,3);
        constrain (this,p3,1,2,1,1,GridBagConstraints.HORIZONTAL,
                GridBagConstraints.CENTER,0.0,0.0,3,3,3,3);
        }
```

```
//-------------------------------------------------------------
        // Event Handler
        public boolean action(Event e,Object arg) {
        if (e.target instanceof Button) {
                answer(Cancel);
                return true;
            }
            else return false;
        }
//-------------------------------------------------------------
    protected void answer(int answer) {
        switch(answer) {
            case Cancel:cancel();break;
        }
    }
//-------------------------------------------------------------
    protected void cancel() {
        this.hide();
        this.dispose();
    }
//-------------------------------------------------------------
//These constrain functions are used to layout the components both
//within thier panels and the panels within the frame.
//(from Java in Nutshell)
    public void constrain (Container container,Component component,
        int grid_x,int grid_y,int grid_width,int grid_height,
        int fill,int anchor,double weight_x,double weight_y,
        int top,int left,int bottom,int right) {

        GridBagConstraints c = new GridBagConstraints();
        c.gridx = grid_x;c.gridy = grid_y;
        c.gridwidth = grid_width;c.gridheight = grid_height;
        c.fill = fill;c.anchor = anchor;
        c.weightx = weight_x;c.weighty = weight_y;
        if(top + bottom + left + right > 0)
            c.insets = new Insets(top,left,bottom,right);
        ((GridBagLayout)container.getLayout()).setConstraints
          (component,c);
        container.add(component);
    }
//-------------------------------------------------------------
    public void constrain (Container container,Component component,
```

```
        int grid_x,int grid_y,int grid_width,int grid_height) {

        constrain (container,component,grid_x,grid_y,grid_width,
        grid_height,GridBagConstraints.NONE,
        GridBagConstraints.NORTHWEST,0.0,0.0,0,0,0,0);
    }
//--------------------------------------------------------------------
    public void constrain (Container container,Component component,
        int grid_x,int grid_y,int grid_width,int grid_height,
        int top,int left,int bottom,int right) {

        constrain (container,component,grid_x,grid_y,grid_width,
        grid_height,GridBagConstraints.NONE,
        GridBagConstraints.NORTHWEST,0.0,0.0,top,left,bottom,right);
    }
}
```

APPENDIX J. REG COMPLETE

```java
// The Reg_Complete class is an class to display registration results
//
// Copyright by:  LT Chris Corgnati for Thesis work with Prof Bhargava
//
// Last modified:  1700  9 July 1997 Version 1.0
//--------------------------------------------------------------------
//--------------------------------------------------------------------
import java.awt.*;

public class Reg_Complete extends Dialog {

    // Control variables are internal to this object
    private static final int Cancel = 1;

    protected Button cancel;

    protected MultiLineLabel label;

    GridBagLayout gridbag = new GridBagLayout();

    // Constructor code
        public Reg_Complete(Frame f,String title,String message,
            String Cancel_label)
        {
        super(f,"DecisionNet",false);
        this.setLayout(new BorderLayout(20,20));
//--------------------------------------------------------------------
// Creation and layout of GUI components
        Panel p1 = new Panel();
        p1.add(label = new MultiLineLabel(message,1));

        Panel p2 = new Panel();
        p2.setLayout(gridbag);

        cancel=new Button (Cancel_label);

        constrain(p2,cancel,1,2,1,1,GridBagConstraints.HORIZONTAL,
            GridBagConstraints.CENTER,0.0,0.0,4,4,4,4);
```

```
            this.show();
//------------------------------------------------------------------
// Arrange the panels
        this.setLayout(gridbag);
// Add panels to the top level window
        constrain (this,p1,1,0,1,1,GridBagConstraints.BOTH,
                GridBagConstraints.CENTER,0.0,0.0,3,3,3,3);
        constrain (this,p2,1,1,1,1,GridBagConstraints.HORIZONTAL,
                GridBagConstraints.CENTER,0.0,0.0,3,3,3,3);
        }
//------------------------------------------------------------------
// Event Handler
        public boolean action(Event e,Object arg) {
        if (e.target instanceof Button) {
                answer(Cancel);
                return true;
            }
            else return false;
        }
//------------------------------------------------------------------
    protected void answer(int answer) {
        switch(answer) {
            case Cancel:cancel();break;
        }
    }
//------------------------------------------------------------------
    protected void cancel() {
        this.hide();
        this.dispose();
    }
//------------------------------------------------------------------
    public void constrain (Container container,Component component,
        int grid_x,int grid_y,int grid_width,int grid_height,
        int fill,int anchor,double weight_x,double weight_y,
        int top,int left,int bottom,int right) {

        GridBagConstraints c = new GridBagConstraints();
        c.gridx = grid_x;c.gridy = grid_y;
        c.gridwidth = grid_width;c.gridheight = grid_height;
        c.fill = fill;c.anchor = anchor;
        c.weightx = weight_x;c.weighty = weight_y;
        if(top + bottom + left + right > 0)
```

104

```java
            c.insets = new Insets(top,left,bottom,right);
        ((GridBagLayout)container.getLayout()).
          setConstraints(component,c);
        container.add(component);
    }
//-----------------------------------------------------------------
    public void constrain (Container container,Component component,
        int grid_x,int grid_y,int grid_width,int grid_height) {

        constrain (container,component,grid_x,grid_y,grid_width,
        grid_height,GridBagConstraints.NONE,
        GridBagConstraints.NORTHWEST,0.0,0.0,0,0,0,0);
    }
//-----------------------------------------------------------------
    public void constrain (Container container,Component component,
        int grid_x,int grid_y,int grid_width,int grid_height,
        int top,int left,int bottom,int right) {

        constrain (container,component,grid_x,grid_y,grid_width,
        grid_height,GridBagConstraints.NONE,
        GridBagConstraints.NORTHWEST,0.0,0.0,top,left,bottom,right);
    }
}
```

APPENDIX K. GREP

```java
// The Grep class is an class to identify lines from taxonomy.db
// that contain a given string arguement
//
// Copyright by:  LT Chris Corgnati for Thesis work with Prof Bhargava
//
// Last modified:  1300  29 July 1997 Version 1.1
//-------------------------------------------------------------------
import java.io.*;
import java.awt.*;

public class Grep {
    List level;

    public Grep(String args,List level,InputStream i) {

        try {
            DataInputStream d = new DataInputStream(i);

            GrepInputStream g = new GrepInputStream(d,args,level);
            String line = null;
            String display = null;

            do {
                //calls GrepInputStream
                line = g.readLine();

                //names in data files start at character 38
                display = line.substring(38);

                //display only values of the attribute in
                //appropriate list
                level.addItem(display);
            }
            while (line != null);
            g.close();
        }

        catch (IOException e) {
            level.addItem("unexpected error");
```

```
        }

        finally {
        }
    }
}
```

APPENDIX L. GREPINPUTSTREAM

```java
// The GrepInputStream class is an object to filter the taxonomy
//
// Copyright by:  LT Chris Corgnati for Thesis work with Prof Bhargava
//
// Last modified:  1300 29 July 1997 Version 1.1
//------------------------------------------------------------------
import java.io.*;
import java.awt.*;

public class GrepInputStream extends FilterInputStream {
    String substring;
    DataInputStream in;
    List level;

    public GrepInputStream(DataInputStream in,String substring,
        List level){
        super(in);
        this.in = in;
        this.substring = substring;
        }
//------------------------------------------------------------------
    public final String readLine() throws IOException {
        String line;
        do {
            line = in.readLine();
        }
        while ((line != null) && line.indexOf(substring) == -1);
        return line;
        }
    }
```

APPENDIX M. QUERYSTRING

```java
// The QueryString class is an class to combine string values for
// input to a CGI script.
//
// Copyright by:  LT Chris Corgnati for Thesis work with Prof Bhargava
//
// Last modified:  1400  30 June 1997 Version 1.0
//------------------------------------------------------------------
import java.net.URLEncoder;

public class QueryString {
   String query;

   public QueryString(Object name,Object value) {
     query = URLEncoder.encode(name.toString()) + "=" +
       URLEncoder.encode(value.toString());
   }

   public void add(Object name,Object value) {
     query += "&" + URLEncoder.encode(name.toString()) + "=" +
       URLEncoder.encode(value.toString());
   }

   public String toString() {
     return query;
   }
}
```

APPENDIX N. MULTILINELABEL

```
// The MultiLineLabel class is an object to parse a long string into
// smaller strings for display.  Java doesn't support a label greater
// than one line.  Borrowed from "Java in a Nutshell".
//
// Copyright by:  LT Chris Corgnati for Thesis work with Prof Bhargava
//
// Last modified:  1300 29 July 1997 Version 1.1
//-------------------------------------------------------------------
import java.awt.*;
import java.util.*;

public class MultiLineLabel extends Canvas {
    public static final int LEFT=0;
    public static final int CENTER=1;
    public static final int RIGHT=2;
    protected String[] lines;
    protected int num_lines;
    protected int margin_width;
    protected int margin_height;
    protected int line_height;
    protected int line_ascent;
    protected int[] line_widths;
    protected int max_width;
    protected int alignment=LEFT;

    protected void newLabel(String label) {
        StringTokenizer t=new StringTokenizer(label,"\n");
        num_lines = t.countTokens();
        lines = new String[num_lines];
        line_widths = new int[num_lines];
        for(int i=0;i<num_lines;i++)lines[i]=t.nextToken();
    }

    protected void measure() {
        FontMetrics fm = this.getFontMetrics(this.getFont());
        if(fm == null) return;

        line_height = fm.getHeight();
        line_ascent = fm.getAscent();
```

```java
        max_width = 0;
        for(int i=0;i<num_lines;i++) {
            line_widths[i] = fm.stringWidth(lines[i]);
            if (line_widths[i]>max_width)max_width=line_widths[i];
        }
    }


    public MultiLineLabel(String label,int margin_width,
        int margin_height,int alignment) {

        newLabel(label);
        this.margin_width = margin_width;
        this.margin_height = margin_height;
        this.alignment = alignment;
    }
    public MultiLineLabel(String label,int margin_width,
        int margin_height) {
        this(label,margin_width,margin_height,LEFT);
    }
    public MultiLineLabel(String label,int alignment) {
        this(label,20,20,alignment);
    }
    public MultiLineLabel(String label) {
        this(label,20,20,LEFT);
    }


    public void setLabel(String label) {
        newLabel(label);
        measure();
        repaint();
    }
    public void setFont(Font f) {
        super.setFont(f);
        measure();
        repaint();
    }
    public void setForeground(Color c) {
        super.setForeground(c);
        repaint();
    }
    public void setAlignment(int a) {alignment=a;repaint();}
    public void setMarginWidth(int mw) {margin_width=mw;repaint();}
```

```java
    public void setMarginHeight(int mh) {margin_height=mh;repaint();}
    public int getAlignment() {return alignment;}
    public int getMarginWidth() {return margin_width;}
    public int getMarginHeight() {return margin_height;}

    public void addNotify() {super.addNotify();measure();}

    public Dimension preferredSize() {
        return new Dimension(max_width + 2*margin_width,
            num_lines * line_height + 2*margin_height);
    }

    public Dimension minimumSize() {
        return new Dimension(max_width,num_lines*line_height);
    }

    public void paint(Graphics g) {
        int x,y;
        Dimension d=this.size();
        y=line_ascent + (d.height - num_lines * line_height)/2;
        for (int i = 0;i < num_lines;i++,y += line_height) {
            switch(alignment) {
            case LEFT:
                x=margin_width;break;
            case CENTER:
            default:
                x=(d.width - line_widths[i]) / 2;break;
            case RIGHT:
                x=d.width - margin_width - line_widths[i];break;
            }
            g.drawString(lines[i],x,y);
        }
    }
}
```

APPENDIX O. INDEXED SEARCH

```
unit Indexed1;    {Technology Search Engine; created by Chris Corgnati.
                   Last updated 23 Jul 97.}

interface

uses
  SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
  Forms, Dialogs, DB, Cgidb, Cgi, DBTables;

type
  TForm1 = class(TForm)
    CGIEnvData1: TCGIEnvData;
    BatchMove1: TBatchMove;
    Table1: TTable;
    Query1: TQuery;
    DataSource1: TDataSource;
    BatchMove2: TBatchMove;
    Table3: TTable;
    DataSource2: TDataSource;
    Query2: TQuery;
    Table2: TTable;
    Query3: TQuery;
    DataSource3: TDataSource;
    Query4: TQuery;
    Table4: TTable;
    DataSource4: TDataSource;
    Table5: TTable;
    BatchMove3: TBatchMove;
    BatchMove4: TBatchMove;
    Table6: TTable;
    BatchMove5: TBatchMove;
    Table7: TTable;
    Table8: TTable;
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
```

```pascal
var
  Form1: TForm1;

implementation

{$R *.DFM}

procedure TForm1.FormCreate(Sender: TObject);

var
  N_R : smallint; {Number of Records in Technolo.db}
  N_T_R : smallint; {Number of Records in Search_Temp_2.db}

  {User selections passed to cgi from applet}
  Var_Value : array [0..4] of string;
  {database field names}
  Category : array [0..4] of string;

  {strings read from applet}
  Temp_Deg_Value : array [0..4,0..2] of string;
  {strings converted to float(single)}
  Deg_Value : array [0..4,0..2] of  Single;

  {Values of variables from temp records}
  temp : array [0..50,0..4] of string;
  {holds provider info for display}
  info : array [0..50,0..2] of string;

  {Record Variable Score}
  R_V_S : array [0..50,0..4] of  Single;
  {Total Score for a given record}
  T_S : array[0..50] of  Single;

  {strings read from applet}
  Temp_Weight : array [0..4] of string;
  {strings converted to float(single)}
  Weight : array [0..4] of Single;

  v : smallint;      {variable counter}
  r : smallint;      {record counter}
  f : smallint;      {field counter}
```

118

```pascal
      j : smallint;      {loop counter}
      i : smallint;
      k : smallint;

      C : string;
      P : string;
      P_Temp : string;
      C_2_Temp : string;
      P_2_Temp : string;

      {control flag for 'all' selections}
      Var_All_Bool : array [0..50,0..4] of Boolean;
      {control flag for exact variable matches}
      Match_Bool : array [0..50,0..4] of Boolean;
      {control flag for child in taxonomy}
      child :  array [0..50,0..4] of Boolean;
      {control flag for second level child in taxonomy}
      Child_Sec_Bool : array [0..50,0..4] of Boolean;
      {control flag for parent in taxonomy}
      parent : array [0..50,0..4] of Boolean;
      {control flag for second level parent in taxonomy}
      Parent_Sec_Bool : array [0..50,0..4] of Boolean;
      {control flag for sibling in taxonomy}
      sibling : array [0..50,0..4] of Boolean;

      test : string;
{-----------------------------------------------------------------------}
   begin
       {Standard cgi stuff}
       with CGIEnvData1 do
            begin
                websiteINIFilename := paramstr(1);
                application.onException := cgiErrorHandler;
                application.processMessages;
                createStdout;
                sendPrologue;

                {find number of records}
                with Table1 do
                  begin
                    Table1.open;
                    N_R := Table1.RecordCount;
```

119

```
                    Table1.close;
                end;

{read in variables from applet and init arrays}
{strings are converted to float or int where appropriate}
            for r := 0 to N_R do
              begin
                for v := 0 to 4 do
                  begin
                    child[r,v] := false;
                    Child_Sec_Bool[r,v] := false;
                    parent[r,v] := false;
                    Parent_Sec_Bool[r,v] := false;
                    sibling[r,v] := false;
                    Var_All_Bool[r,v] := false;
                  end;
              end;

        {used to generate the queries}
        C := ('Child');
        P := ('Parent');

        {correspond to column headings in Technolo.db}
        Category[0] := ('tProblemArea');
        Category[1] := ('tFunctionalArea');
        Category[2] := ('tSolutionMethod');
        Category[3] := ('tIndType');
        Category[4] := ('tOrgType');

        {user selected search criteria-from applet}
        Var_Value[0] := GetSmallField('tProblemArea');
        Var_Value[1] := GetSmallField('tFunctionalArea');
        Var_Value[2] := GetSmallField('tSolutionMethod');
        Var_Value[3] := GetSmallField('tIndType');
        Var_Value[4] := GetSmallField('tOrgType');

        {user selected weights and degredation
        multipliers-from applet}
        Temp_Weight[0] := GetSmallField('Problem_Weight');
        Weight[0] := StrToFloat(Temp_Weight[0]);
        Temp_Deg_Value[0,0] := GetSmallField('Problem_C_Deg');
        Deg_Value[0,0] := StrToFloat(Temp_Deg_Value[0,0]);
```

```
Temp_Deg_Value[0,1] := GetSmallField('Problem_P_Deg');
Deg_Value[0,1] := StrToFloat(Temp_Deg_Value[0,1]);
Temp_Deg_Value[0,2] := GetSmallField('Problem_S_Deg');
Deg_Value[0,2] := StrToFloat(Temp_Deg_Value[0,2]);

Temp_Weight[1] := GetSmallField('Functional_Weight');
Weight[1] := StrToFloat(Temp_Weight[1]);
Temp_Deg_Value[1,0] := GetSmallField('Functional_C_Deg');
Deg_Value[1,0] := StrToFloat(Temp_Deg_Value[1,0]);
Temp_Deg_Value[1,1] := GetSmallField('Functional_P_Deg');
Deg_Value[1,1] := StrToFloat(Temp_Deg_Value[1,1]);
Temp_Deg_Value[1,2] := GetSmallField('Functional_S_Deg');
Deg_Value[1,2] := StrToFloat(Temp_Deg_Value[1,2]);

Temp_weight[2] := GetSmallField('Solution_Weight');
Weight[2] := StrToFloat(Temp_Weight[2]);
Temp_Deg_Value[2,0] := GetSmallField('Solution_C_Deg');
Deg_Value[2,0] := StrToFloat(Temp_Deg_Value[2,0]);
Temp_Deg_Value[2,1] := GetSmallField('Solution_P_Deg');
Deg_Value[2,1] := StrToFloat(Temp_Deg_Value[2,1]);
Temp_Deg_Value[2,2] := GetSmallField('Solution_S_Deg');
Deg_Value[2,2] := StrToFloat(Temp_Deg_Value[2,2]);

Temp_Weight[3] := GetSmallField('Industry_Weight');
Weight[3] := StrToFloat(Temp_Weight[3]);
Temp_Deg_Value[3,0] := GetSmallField('Industry_C_Deg');
Deg_Value[3,0] := StrToFloat(Temp_Deg_Value[3,0]);
Temp_Deg_Value[3,1] := GetSmallField('Industry_P_Deg');
Deg_Value[3,1] := StrToFloat(Temp_Deg_Value[3,1]);
Temp_Deg_Value[3,2] := GetSmallField('Industry_S_Deg');
Deg_Value[3,2] := StrToFloat(Temp_Deg_Value[3,2]);

Temp_Weight[4] := GetSmallField('Organization_Weight');
Weight[4] := StrToFloat(Temp_Weight[4]);
Temp_Deg_Value[4,0] := GetSmallField('Organization_C_Deg');
Deg_Value[4,0] := StrToFloat(Temp_Deg_Value[4,0]);
Temp_Deg_Value[4,1] := GetSmallField('Organization_P_Deg');
Deg_Value[4,1] := StrToFloat(Temp_Deg_Value[4,1]);
Temp_Deg_Value[4,2] := GetSmallField('Organization_S_Deg');
Deg_Value[4,2] := StrToFloat(Temp_Deg_Value[4,2]);
{-------------------------------------------------------------------}
{quick search for technologies that may be of interest
```

121

```
         send from Technolo.db to search_temp.db}

      {empty Search_Temp.db}
         with Table2 do
           begin
             Active := False;
             DatabaseName := 'DNET';
             TableName := 'Search_Temp';
             TableType := ttParadox;
             EmptyTable;
           end;

     {select technologies that are potentially useful and send them
      to search_temp.db}
      with Query1 do
        begin
          for v := 0 to 4 do
            begin
              Query1.SQL.clear;
              Query1.SQL.add ('SELECT * FROM TECHNOLO');
              Query1.SQL.add ('WHERE ('+ Category[v] + ' = "'
+ Var_Value[v] + '") ');
              Query1.SQL.add ('ORDER BY ProviderID,TechID');
              BatchMove1.Execute;
              Query1.close;
            end;
          end;

      {empty Search_Temp_2.db}
      with Table3 do
        begin
          Active := False;
          DatabaseName := 'DNET';
          TableName := 'Search_Temp_2';
          TableType := ttParadox;
          EmptyTable;
        end;

     {remove duplicate records from Search_Temp.db and copy results
      to Search_Temp_2.db}
       with Table2 do
         begin
```

122

```
                Query2.SQL.clear;
                Query2.SQL.add ('SELECT DISTINCT ProviderID,TechID,
        TechName,tProblemArea,tFunctionalArea,
        tSolutionMethod,tIndType,tOrgType ');
                Query2.SQL.add (' FROM Search_Temp ');
                Query2.SQL.add ('ORDER BY ProviderID');
                BatchMove2.Execute;
                Query2.close;
            end;
{----------------------------------------------------------------------}
{go through every record in Search_Temp_2.db and evaluate its value
against the user selections.  The relevence score should be sent to
R_V_S[r,v] for later use in calculating T_S}

with Table3 do
   {find the number of records in Search_Temp_2.db}
   begin
     Table3.open;
     N_T_R := Table3.RecordCount;
     Table3.close;

     {initialize arrays}
     C_2_Temp := (' ');
     P_2_Temp := (' ');
     for r := 0 to N_T_R do
       begin
         for v := 0 to 4 do
           begin
             R_V_S[r,v] := (0.0);
             info[r,v] := (' ');
             temp[r,v] := (' ');
           end;
         T_S[r] := (0.0);
       end;
   end;

with Table3 do
   begin
     Table3.open;
     Table3.First();
     {read in one record at a time to analyze the variables}
     for r := 0 to (N_T_R - 1) do
```

123

```
begin
  {3 to 7 are actual field #s in db}
  temp[r,0] := Table3.Fields[3].AsString;
  temp[r,1] := Table3.Fields[4].AsString;
  temp[r,2] := Table3.Fields[5].AsString;
  temp[r,3] := Table3.Fields[6].AsString;
  temp[r,4] := Table3.Fields[7].AsString;

  info[r,0] := Table3.Fields[0].AsString;
  info[r,1] := Table3.Fields[1].AsString;
  info[r,2] := Table3.Fields[2].AsString;

  Table3.Next();
  end;
  Table3.close;
end;


{check for relationships and turn on boolean flags once a
relationship has been established.  Order of precedence:
user selected All, exact variable match, child, sibling,
parent, second level child, second level parent}


with Query4 do
  begin
    for r := 0 to (N_T_R - 1) do
      begin
        for v := 0 to 4 do
          begin
            {if user selects ALL we consider it a match}
            if (Var_Value[v] = 'ALL') then
              begin
                Var_All_Bool[r,v] := True;
              end;

            {a match is assigned a score of 1.0}
            if (Var_Value[v] = temp[r,v]) then
              begin
                Match_Bool[r,v] := True;
              end;

            {check for child relationships}
```

```
if (Var_All_Bool[r,v] <> True) and
   (Match_Bool[r,v] <> True) then
  begin
    with Table5 do
      begin
        Active := False;
        DatabaseName := 'DNET';
        TableName := 'Search_Temp_3';
        TableType := ttParadox;
        EmptyTable;
      end;
    Query4.SQL.clear;
    Query4.SQL.add ('SELECT * FROM Taxonomy');
    Query4.SQL.add ('WHERE ('+C+' = "'+Var_Value[v]
              +'" AND '+P+' = "'+temp[r,v]+ '")');
    BatchMove3.Execute;
    Query4.close;
    with Table5 do
      begin
        Table5.open;
        if Table5.RecordCount > 0 then
          begin
            child[r,v] := True;
          end;
        Table5.close;
      end;
  end;

{check for parent relationship}
if (Var_All_Bool[r,v] <> True) and
   (Match_Bool[r,v] <> True) and
   (child[r,v] <> True) then
  begin
    with Table5 do
      begin
        Active := False;
        DatabaseName := 'DNET';
        TableName := 'Search_Temp_3';
        TableType := ttParadox;
        EmptyTable;
      end;
    Query4.SQL.clear;
```

```
                    Query4.SQL.add ('SELECT * FROM Taxonomy');
                    Query4.SQL.add ('WHERE ('+C+' = "'+temp[r,v]+'" AND '
+P+' = "'+Var_Value[v]+ '")');
                    BatchMove3.Execute;
                    Query4.close;
                    with Table5 do
                      begin
                        Table5.open;
                        if Table5.RecordCount > 0 then
                          begin
                            parent[r,v] := True;
                          end;
                        Table5.close;
                      end;
                 end;

               {check for sibling relationship}
               if (Var_All_Bool[r,v] <> True) and
                  (Match_Bool[r,v] <> True) and
                  (child[r,v] <> True) and
                  (parent[r,v] <> True) then
                 begin
                  with Table5 do
                    begin
                      Active := False;
                      DatabaseName := 'DNET';
                      TableName := 'Search_Temp_3';
                      TableType := ttParadox;
                      EmptyTable;
                    end;
                  Query4.SQL.clear;
                  Query4.SQL.add ('SELECT * FROM Taxonomy');
                  Query4.SQL.add ('WHERE ('+C+' = "'+Var_Value[v]+'")');
                  BatchMove3.Execute;
                  Query4.close;

                   with Table5 do
                     begin
                       Table5.open;
                       Table5.First();
                       P_Temp := Table5.Fields[0].AsString;
                       Table5.Close();
```

126

```
                        {check to see if Parent is on top level
                        -that is a meaningless relationship}
                        if (P_Temp <> 'Problem Area') and
(P_Temp <> 'Functional Area') and
                           (P_Temp <> 'Solution Method') and
(P_Temp <> 'Industry Type') and
                              (P_Temp <> 'Organization Type') then
                           begin
                             Query4.SQL.clear;
                             Query4.SQL.add ('SELECT * FROM Taxonomy');
                             Query4.SQL.add ('WHERE ('+C+' = "'+temp[r,v]
                                     +'"AND '+P+' = "'+P_Temp+ '")');
                             BatchMove3.Execute;
                             Query4.close;

                             Table5.open;
                             k := Table5.RecordCount;
                             Table5.Close();

                             if (k > 0) then
                               begin
                                 sibling[r,v] := True;
                               end;
                           end;
                     end;
                end;

                {check for second level child relationship}
                if (Var_All_Bool[r,v] <> True) and
   (Match_Bool[r,v] <> True) and
                   (child[r,v] <> True) and
   (parent[r,v] <> True) and
                   (sibling[r,v] <> True) then
                begin
                  with Table6 do
                    begin
                      Active := False;
                      DatabaseName := 'DNET';
                      TableName := 'Tax_Temp';
                      TableType := ttParadox;
                      EmptyTable;
```

```
              end;

          Query4.SQL.clear;
          Query4.SQL.add ('SELECT * FROM Taxonomy');
          Query4.SQL.add ('WHERE ('+C+' = "'
                          +Var_Value[v]+'")');
          BatchMove4.Execute;
          Query4.close;

          with Table6 do
            Table6.open;
            Table6.First();
            begin
              j := Table6.RecordCount;
              for i := 0 to (j) do
                begin
                  C_2_Temp := Table6.Fields[0].AsString;
                  if (C_2_Temp <> 'Problem Area') and
      (C_2_Temp <> 'Functional Area') and
                      (C_2_Temp <> 'Solution Method') and
              (C_2_Temp <> 'Industry Type') and
                      (C_2_Temp <> 'Organization Type') then

                    begin
                      Query4.SQL.clear;
                      Query4.SQL.add ('SELECT * FROM Taxonomy');
                      Query4.SQL.add ('WHERE ('+P+' = "'+temp[r,v]
  +'" AND '+C+' = "'+C_2_Temp+ '")');
                      BatchMove5.Execute;
                      Query4.close;

                      with Table7 do
                        begin
                          Table7.open;
                            k := Table7.RecordCount;
                            Table7.Close();

                          if (k > 0) then
                            begin
                              Child_Sec_Bool[r,v] := True;
                            end;
                      end;
```

128

```
                        end;
                    Table6.Next();
                        end;
                  end;
                Table6.Close();
            end;

            {check for second level parent relationship}
            if (Var_All_Bool[r,v] <> True) and
(Match_Bool[r,v] <> True) and
                (child[r,v] <> True) and
(parent[r,v] <> True) and
                (sibling[r,v] <> True) and
                (Child_Sec_Bool[r,v] <> True) then

            begin
              with Table6 do
                begin
                  Active := False;
                  DatabaseName := 'DNET';
                  TableName := 'Tax_Temp';
                  TableType := ttParadox;
                  EmptyTable;
                end;

              Query4.SQL.clear;
              Query4.SQL.add ('SELECT * FROM Taxonomy');
              Query4.SQL.add ('WHERE ('+P+' = "'
                            +Var_Value[v]+'")');
              BatchMove4.Execute;
              Query4.close;

              with Table6 do
                Table6.open;
                Table6.First();
                begin
                  j := Table6.RecordCount;

                  for i := 0 to j do
                    begin
                      P_2_Temp := Table6.Fields[1].AsString;
                      Query4.SQL.clear;
```

```
                        Query4.SQL.add ('SELECT * FROM Taxonomy');
                        Query4.SQL.add ('WHERE ('+P+' = "'+P_2_Temp+'"
        AND '+C+' = "'+temp[r,v]+ '")');
                        BatchMove5.Execute;
                        Query4.close;

                        with Table7 do
                          begin
                            Table7.open;
                               k := Table7.RecordCount;
                            Table7.Close();

                          if k > 0 then
                            begin
                              Parent_Sec_Bool[r,v] := True;
                            end;
                          end;
                        Table6.Next();
                        end;
                    end;
                  Table6.Close();
              end;
            end;
        end;
      end;
{----------------------------------------------------------------------}
{evaluate booleans and set the R_V_S accordingly}

for r := 0 to (N_T_R - 1) do
  begin
    for v := 0 to 4 do
      begin
        {if user selects ALL we consider it a match}
        if (Var_All_Bool[r,v] = True) then
          begin
            R_V_S[r,v] := 1.0
          end

        {a match is assigned a score of 1.0}
        else if (Match_Bool[r,v] = True) then
          begin
            R_V_S[r,v] := 1.0
```

130

```
        end

{query taxonomy.db for child relationship}
else if (child[r,v] = True) then
  begin
    R_V_S[r,v] := Deg_Value[v,0]
  end

{query taxonomy.db for parent relationship}
else if (parent[r,v] = True) then
  begin
    R_V_S[r,v] := Deg_Value[v,1]
  end

{query taxonomy.db for sibling relationship}
else if (sibling[r,v] = True) then
  begin
    R_V_S[r,v] := Deg_Value[v,2]
  end

{query taxonomy.db for second level child relationship}
else if (Child_Sec_Bool[r,v] = True) then
  begin
    R_V_S[r,v] := Deg_Value[v,0]*Deg_Value[v,0]
  end

{query taxonomy.db for second level parent relationship}
else if (Parent_Sec_Bool[r,v] = True) then
  begin
    R_V_S[r,v] := Deg_Value[v,1]*Deg_Value[v,1]
  end

{if technology is listed as all we assign a score of 0.5}
else if (temp[r,v] = 'ALL' ) then
  begin
    R_V_S[r,v] := 0.5
  end

{no useful result yields a score of 0.0}
else
  begin
    R_V_S[r,v] := 0.0;
```

```
            end;
        end;
    end;


{----------------------------------------------------------------------}
{Calculate the Total Score for each record in Search_Temp_2.db}

for r := 0 to (N_T_R - 1) do    {should use RecordCount}
  begin
    T_S[r] := ((Weight[0])*(R_V_S[r,0])) +
              ((Weight[1])*(R_V_S[r,1])) +
              ((Weight[2])*(R_V_S[r,2])) +
              ((Weight[3])*(R_V_S[r,3])) +
              ((Weight[4])*(R_V_S[r,4]));
  end;
{----------------------------------------------------------------------}
{standard header information}
{all instances of the character & are for output parsing and
have no effect on the search}
send( '                    DecisionNet Search Engine Results' + '&' );
send( ' '+'&' );

{user selection information}
send('User Selections:'+ '&');
send('Problem Area  = ' + Var_Value[0]+ '&');
send('Functional Area = ' + Var_Value[1]+ '&');
send('Solution Method = ' + Var_Value[2]+ '&');
send('Industry Type = ' + Var_Value[3]+ '&');
send('Organization Type = ' + Var_Value[4]+ '&');
send(' '+ '&');

send('Problem Area Weight = ' + FloatToStr(Weight[0]) + '&');
send('Functional Area Weight = ' + FloatToStr(Weight[1]) + '&');
send('Solution Method Weight = ' + FloatToStr(Weight[2]) + '&');
send('Industry Type Weight = ' + FloatToStr(Weight[3]) + '&');
send('Organization Type Weight = ' + FloatToStr(Weight[4]) + '&');
send(' '+ '&');

{sends T_S and R_V_S array values for individual technologies}

for r := 0 to (N_T_R - 1) do
  begin
```

132

```
        send('TechID = ' + info[r,1] + '    TechName = ' + info[r,2]+ '&');
        send(' '+ '&');
        send('Relevency Score = ' + FloatToStr(T_S[r])+ '&');
        send('Score for (Problem Area) =
            ' + FloatToStr(R_V_S[r,v])+ '&');
        send('Score for (Functional Area) =
            ' + FloatToStr(R_V_S[r,v])+ '&');
        send('Score for (Solution Method) =
            ' + FloatToStr(R_V_S[r,v])+ '&');
        send('Score for (Industry Type) =
            ' + FloatToStr(R_V_S[r,v])+ '&');
        send('Score for (Organization Type) =
            ' + FloatToStr(R_V_S[r,v])+ '&');
        send(' '+ '&');
    end;


    {standard footer information}
    send( 'This search engine was created by LT Chris Corgnati'+ '&');
    send( 'for Professor Hemant Bhargava at the Naval Postgraduate
        School.'+ '&' );
    send( 'Generated on ' + webdate(now) + '&');

closeStdout;
closeApp( application );
        end;
    end;
end.
```

APPENDIX P. TAXONOMY LEVEL 1

Object Type	ALL
Object Type	Algorithm
Object Type	Data Set
Object Type	Decision Support System
Object Type	Example
Object Type	Model Schema
Object Type	Modeling Language
Object Type	Solver
Problem Area	ALL
Problem Area	Asset Pricing
Problem Area	Assignment
Problem Area	Bin packing
Problem Area	Capital budgeting
Problem Area	Communications networks
Problem Area	Corporate strategy
Problem Area	Cost analysis
Problem Area	Crew scheduling
Problem Area	Depreciation
Problem Area	Environment systems analysis
Problem Area	Facilities/equipment planning
Problem Area	Fire models
Problem Area	Hierarchical production planning
Problem Area	Inventory
Problem Area	Investment
Problem Area	Knapsack problem
Problem Area	Job shop scheduling
Problem Area	Layout/Location of equipment
Problem Area	Learning/Training
Problem Area	Lifecycle
Problem Area	Location analysis
Problem Area	Maintenance/Repair
Problem Area	Manpower planning
Problem Area	Material Handling
Problem Area	Planning
Problem Area	Production
Problem Area	Quality Control
Problem Area	Reliability of systems
Problem Area	Risk assessment/management
Problem Area	Safety

Problem Area	Scheduling/sequencing
Problem Area	Taxation
Problem Area	Utility/Preferences
Problem Area	Vehicle routing
Problem Area	Yield management
Functional Area	ALL
Functional Area	Administration
Functional Area	Engineering
Functional Area	Finance
Functional Area	Legal
Functional Area	Logistics
Functional Area	Maintenance
Functional Area	Marketing
Functional Area	Payroll
Functional Area	Personnel
Functional Area	Shopping
Functional Area	Supply
Functional Area	Testing
Functional Area	Training
Functional Area	Transportation
Functional Area	Telecommunications
Industry Type	ALL
Industry Type	Aerospace/Space
Industry Type	Accounting
Industry Type	Agriculture
Industry Type	Arts
Industry Type	Business/Commerce
Industry Type	Communications
Industry Type	Construction
Industry Type	Consumer
Industry Type	Economics
Industry Type	Education
Industry Type	Emergency Services
Industry Type	Energy
Industry Type	Engineering
Industry Type	Entertainment/Media
Industry Type	Environment/Ecology
Industry Type	Finance
Industry Type	Government
Industry Type	Health/Medicine
Industry Type	International Trade
Industry Type	Labor

Industry Type	Law
Industry Type	Libraries
Industry Type	Manufacturing
Industry Type	Marketing
Industry Type	Military
Industry Type	Natural Resources
Industry Type	Petro-Chemical
Industry Type	Population
Industry Type	Recreation and Sports
Industry Type	Scientific Research
Industry Type	Transportation
Organization Type	ALL
Organization Type	Military
Organization Type	Government(non-military)
Organization Type	Commercial
Organization Type	Non-commercial/non-profit
Organization Type	Personal/Individual
Solution Area	ALL
Solution Area	Simulation/stochastic modeling
Solution Area	Statistics/probability
Solution Area	Gaming
Solution Area	Linear Programming
Solution Area	Optimization
Solution Area	Numerical Math
Solution Area	Spreadsheet modeling
Solution Area	Knowledge based
Solution Area	Symbolic Math(calculus)
Solution Area	Decision Analysis
Solution Area	Other

APPENDIX Q. TAXONOMY LEVEL 2

Agriculture	Crops
Agriculture	Food Production
Agriculture	Livestock
Agriculture	Weather
Arts	Fine Arts
Arts	Music
Arts	Theater
Business/Commerce	Apparel
Business/Commerce	General Retailing
Business/Commerce	Mail-order
Business/Commerce	Restaurant
Business/Commerce	Wholesale
Construction	Architecture
Construction	Materials
Economics	Macroeconomics
Economics	Microeconomics
Education	K-12
Education	Undergraduate
Education	Graduate
Education	Career training
Education	Vocational
Emergency Services	Fire
Emergency Services	Police
Emergency Services	Hospitals
Entertainment/Media	Broadcasting
Entertainment/Media	Film
Entertainment/Media	Music
Entertainment/Media	Publishing
Entertainment/Media	Radio
Entertainment/Media	Television
Environment/Ecology	Air quality
Environment/Ecology	Forestry
Environment/Ecology	Pollution
Environment/Ecology	Recycling
Environment/Ecology	Water quality
Finance	Banking
Finance	Insurance
Finance	Investment
Finance	Mortgage

Finance	Personal
Finance	Real Estate
Government	Elections
Government	Energy Policy
Government	Public Works
Government	Regulations
Government	Tax Policy
Government	Urban Systems
Health/Medicine	Exercise
Health/Medicine	Health care
Health/Medicine	Nutrition
Libraries	Archives
Manufacturing	Aerospace
Manufacturing	Automotive
Manufacturing	Building materials
Manufacturing	Chemical
Manufacturing	Clothing/Textiles
Manufacturing	Forest/Paper products
Manufacturing	Furniture
Manufacturing	Metals
Manufacturing	Pharmaceuticals
Manufacturing	Plastics
Military	Avionics
Military	Command and Control
Military	Data processing
Military	Simulation
Military	Software tools
Military	Telecommunications
Military	Testing
Scientific Research	Chemistry
Scientific Research	Geography
Scientific Research	Engineering
Scientific Research	Mathematics
Scientific Research	Meteorology
Scientific Research	Oceanography
Scientific Research	Physics
Scientific Research	Statistics
Transportation	Airline
Transportation	Automobile
Transportation	Cargo
Transportation	Railroad
Transportation	Shipping

Simulation/stochastic modeling	Simulation
Simulation/stochastic modeling	Queuing
Simulation/stochastic modeling	Reliability
Statistics/probability	Approximation
Statistics/probability	Statistics,Probability
Optimization	Linear Algebra
Optimization	Optimization
Numerical Math	Arithmetic
Numerical Math	Number theory
Numerical Math	Elementary/special functions
Numerical Math	Interpolation
Numerical Math	Nonlinear equations
Numerical Math	Approximation
Symbolic Math(calculus)	Elementary/Special Functions
Symbolic Math(calculus)	Differentiation/integration
Symbolic Math(calculus)	Differential/integral equations
Symbolic Math(calculus)	Integral transforms
Decision Analysis	Decision Trees
Decision Analysis	Utility Theory
Decision Analysis	Influence Diagrams
Decision Analysis	AHP
Decision Analysis	Risk Analysis

APPENDIX R. TAXONOMY LEVEL 3

This file has been formatted for display. Where line
breaks appear, it is only to fit the text on this page. The
actual data file has no blank lines and the child variable
always starts at character position 39.

Simulation	Discrete
Simulation	Continuous(Markov)
Reliability	Quality control
Reliability	Electrical network
Approximation	Least squares approximation
Approximation	Minimax approximation
Approximation	Least absolute value approximation
Approximation	Other approximations
Approximation	Smoothing
Approximation	Service routines,approximation
Approximation	Grid or knot generation
Approximation	Manipulation of basis functions
Approximation	Other
Statistics,Probability	Data Summarization
Statistics,Probability	Data manipulation
Statistics,Probability	Elementary statistical graphics
Statistics,Probability	Elementary data analysis
Statistics,Probability	Function evaluation
Statistics,Probability	Random number generation
Statistics,Probability	Analysis of variance
Statistics,Probability	Regression
Statistics,Probability	Categorical data analysis
Statistics,Probability	Time series analysis
Statistics,Probability	Correlation analysis
Statistics,Probability	Discriminant analysis
Statistics,Probability	Covariance structure models
Statistics,Probability	Cluster analysis
Statistics,Probability	Life testing,survival analysis
Statistics,Probability	Random number generation
Statistics,Probability	Multidimensional scaling
Statistics,Probability	Statistical data sets
Linear Algebra	Systems of linear equations
Linear Algebra	Determinants

Linear Algebra	Eigenvalues,eigenvectors"
Linear Algebra	QR decomposition/Gram-Schmidt orthogonalization
Linear Algebra	Singular value decomposition
Linear Algebra	Update matrix decomposition
Linear Algebra	Other matrix equations
Linear Algebra	Generalized inverses
Optimization	Unconstrained
Optimization	Constrained
Optimization	Optimal Control
Optimization	Service routines
Arithmetic	Integer
Arithmetic	Rational
Arithmetic	Real
Arithmetic	Complex
Arithmetic	Interval
Arithmetic	Change of representation
Arithmetic	Sequences
Elementary/special functions	Integer-valued functions
Elementary/special functions	Powers,roots,reciprocals
Elementary/special functions	Polynomials
Elementary/special functions	Elementary transcendental functions
Elementary/special functions	Exponential/logarithmic integrals
Elementary/special functions	Gamma
Elementary/special functions	Error functions
Elementary/special functions	Legendre functions
Elementary/special functions	Bessel functions
Elementary/special functions	Confluent hypergeometric functions
Elementary/special functions	Coulomb wave functions
Elementary/special functions	Jacobian elliptic functions/theta
Elementary/special functions	Elliptical integrals
Elementary/special functions	Weierstrass elliptic functions
Elementary/special functions	Parabolic cylinder functions
Elementary/special functions	Mathieu functions
Elementary/special functions	Spheroidal wave functions
Elementary/special functions	Other special functions
Interpolation	Univariate data
Interpolation	Multivariate data

Interpolation	Service routines, interpolation
Nonlinear equations	Single equation
Nonlinear equations	System of equations
Nonlinear equations	Service routine
Approximation	Least squares approximation
Approximation	Minimax approximation
Approximation	Least absolute value approximation
Approximation	Other analytic approximations
Approximation	Smoothing
Approximation	Service routines, approximation
Approximation	Grid or Knot generation
Approximation	Manipulation of basis functions
Approximation	Other
Approximation	Nonlinear least squares
Elementary/special functions	Integer valued functions
Elementary/special functions	Powers, roots, reciprical
Elementary/special functions	Polynomials
Differentiation, integration	Multidimensional integrals
Differentiation, integration	Service routines
Differential/integral equations	Ordinary differential equations
Differential/integral equations	Partial differential equations
Differential/integral equations	Integral equations
Integral transforms	Trigonometric/fast Fourier transforms
Integral transforms	Convolutions
Integral transforms	Laplace transforms
Integral transforms	Hilbert transforms

APPENDIX S. TAXONOMY LEVEL 4

This file has been formatted for display. Where line
breaks appear, it is only to fit the text on this page. The
actual data file has no blank lines and the child variable
always starts at character position 39.

Least squares approximation	Linear least squares
Least squares approximation	Nonlinear least squares
Service routines for approx	Evaluation of fitted functions
Data Summarization	One-dimensional data
Data Summarization	Two-dimensional data
Data Summarization	Multi-dimensional data
Data manipulation	Transform
Data manipulation	Tally
Data manipulation	Subset
Data manipulation	Merge
Data manipulation	Construct new variables
Elementary statistical graphics	One-dimensional data
Elementary statistical graphics	Two-dimensional data
Elementary statistical graphics	Three-dimensional data
Elementary statistical graphics	Multi-dimensional data
Elementary data analysis	One-dimensional data
Elementary data analysis	Two-dimensional data
Elementary data analysis	Multi-dimensional data
Elementary data analysis	Multiple Multi-dimensional data
Function evaluation	Univariate
Function evaluation	Multivariate
Random number generation	Univariate
Random number generation	Multivariate
Random number generation	Service routines
Analysis of variance	One-way
Analysis of variance	Two-way
Analysis of variance	Three-way
Analysis of variance	Multi-way
Analysis of variance	Multivariate
Analysis of variance	Generate experimental designs
Analysis of variance	Service routines
Regression	Simple linear
Regression	Polynomial
Regression	Multiple linear

147

Regression	Polynomial in several variables
Regression	Nonlinear
Regression	Simultaneous
Regression	Spline
Regression	EDA
Regression	Service routines
Categorical data analysis	2 by 2 tables
Categorical data analysis	Two-way tables
Categorical data analysis	Log-linear model
Categorical data analysis	EDA
Time series analysis	Univariate
Time series analysis	Two time series
Cluster analysis	One-way
Cluster analysis	Two-way
Cluster analysis	Display
Cluster analysis	Service routines
Systems of linear equations	Real nonsymmetric matrices
Systems of linear equations	Real symmetric matrices
Systems of linear equations	Complex non-Hermitian matrices
Systems of linear equations	Complex Hermitian matrices
Systems of linear equations	Associated operations
Determinants	Real nonsymmetric matrices
Determinants	Real symmetric matrices
Determinants	Complex non-Hermitian matrices
Determinants	Complex Hermitian matrices
Eigenvalues,eigenvectors	Ordinary eigenvalue problems
Eigenvalues,eigenvectors	Generalized eigenvalue problems
Eigenvalues,eigenvectors	Associated operations
Update matrix decomposition	LU
Update matrix decomposition	Cholesky
Update matrix decomposition	QR
Update matrix decomposition	Singular value
Generalized inverses	Unconstrained
Generalized inverses	Constrained
Generalized inverses	Generalized inverses
Unconstrained	Univariate
Unconstrained	Multivariate
Constrained	Linear programming
Constrained	Transportation/assignment
Constrained	Integer programming
Constrained	Network
Constrained	Quadratic programming

148

Constrained	Geometric programming
Constrained	Dynamic programming
Constrained	General nonlinear programming
Constrained	Problem input
Service routines	Problem scaling
Service routines	Check user-supplied derivatives
Service routines	Find feasible point
Service routines	Check for redundancy
Service routines	Other
Real	Standard precision
Real	Extended precision
Real	Extended range
Complex	Standard precision
Complex	Extended precision
Complex	Extended range
Change of representation	Base conversion
Change of representation	Decomposition,construction
Polynomials	Orthoginal
Polynomials	Non-orthogonal
Elementary transcendental functions	Trigonometric,inverse trigonometric
Elementary transcendental functions	Exponential, logarithmic
Elementary transcendental functions	Hyperbolic, inverse hyperbolic
Elementary transcendental functions	Integrals
Gamma	Gamma/log gamma/reciprocal gamma
Gamma	Beta/log beta
Gamma	Psi functions
Gamma	Polygamma functions
Gamma	Incomplete gamma
Gamma	Incomplete beta
Gamma	Riemann zeta
Error functions	Functions/inverses/integral/ distributions
Error functions	Fresnel integrals
Error functions	Dawson's integral
Bessel functions	J/Y/H_1/H_2
Bessel functions	I/K
Bessel functions	Kelvin functions
Bessel functions	Airy and Scorer
Bessel functions	Stuve/Anger/Weber
Bessel functions	Integrals of Bessel
Univariate data	Polynomial splines

Univariate data	Polynomials
Univariate data	Other functions
Multivariate data	Gridded
Multivariate data	Scattered
Service routines,interpolation	Evaluate fitted functions
Service routines,interpolation	Grid or knot generation
Service routines,interpolation	Manipulation of basis functions
Service routines,interpolation	Other
Single equation	Polynomial
Single equation	Nonpolynomial
Least squares approximation	Linear least squares
Least squares approximation	Nonlinear least squares
Service routines,approximation	Evaluation of fitted functions
Other	Unconstrained
Other	Constrained
Nonlinear least squares	Unconstrained
Nonlinear least squares	Constrained
Polynomials	Orthogonal
Polynomials	Non-orthogonal on grid
Multidimensional integrals	hyper-rectangular regions
Multidimensional integrals	n-dimensional quadrature
Ordinary differential equations	Initial value
Ordinary differential equations	Multipoint boundary value
Ordinary differential equations	Service routines
Partial differential equations	Initial boundary value
Partial differential equations	Elliptic boundary value
Trigonometric/fast Fourier transforms	One-dimensional
Trigonometric/fast Fourier transforms	Multidimensional

LIST OF REFERENCES

[1] Library of Congress. Library of Congress, 1997. [http://rs7tr0.loc.gov/].

[2] Frankfurt University. The Java Repository, 1997. [http:// java.wiwi.uni.frankurt.de:8080/java/owa/jr_search].

[3] H. Bhargava. Welcome to Decisionnet, 1997. [http://dnet.sm.nps.navy.mil/].

[4] Excite Inc. Excite Search, 1997. [http://www.excite.com].

[5] Lycos Inc. Lycos, Your Personal Internet Guide, 1997. [http://www.lycos.com].

[6] H. Bhargava, A. King, and D. McQuay. Decisionnet: An Architecture for Modeling and Decision Support over the World Wide Web. In Tung X. Bui, editor, *Proceedings of the Third International Society for Decision Support Systems Conference, Vol. II*, pages 541–550, Hong Kong, 1995. International Society for DSS.

[7] S.H. Earley. Decisionnet: A Database Approach. Master's thesis, Naval Postgraduate School, 1996.

[8] P.M. Rogers. Indexing and Retrieval in Digital Libraries: Developing Taxonomies for a Repository of Decision Technologies. Master's thesis, Naval Postgraduate School, 1996.

[9] R. Boisvert. The Guide to Available Mathematical Software Problem Classification System. *Communications in Statistics, part B-Simulation and Computation*, 20(4), 1991.

[10] D. Flanagan. *Java in a Nutshell.* OReilly and Associates, 1996.

[11] A. Taylor. *JDBC Developer's Resource.* Prentice Hall PTR, 1997.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center 2
   8725 John J. Kingman Rd., Ste 0944
   Ft. Belvoir, VA 22060-6218

2. Dudley Knox Library 2
   Naval Postgraduate School
   411 Dyer Rd.
   Monterey, CA 93943-5101

3. Professor Hemant Bhargava (Code SM/BH) 3
   Systems Management Departement
   Naval Postgraduate School
   Monterey, CA 93940-5000

4. Professor Gordon H. Bradley (Code OR/BZ) 1
   Department of Operations Research
   Naval Postgraduate School
   Monterey, CA 93943-5000

5. Professor Carl Jones (Code SM/CJ) 1
   Systems Management Departement
   Naval Postgraduate School
   Monterey, CA 93943-5000

6. Professor Ramayya Krishnan 1
   The Heinz School
   Carnegie Mellon University
   Pittsburgh, PA 15213

7. Professor Rudolf Müller 1
   Institüt für Wirtschaftsinformatik
   Humboldt-Universität zu Berlin
   Spandauer Str. 1
   10178 Berlin, Germany

8. LT Christopher M. Corgnati 2
   27 Middleton Place
   Hertford NC 27944